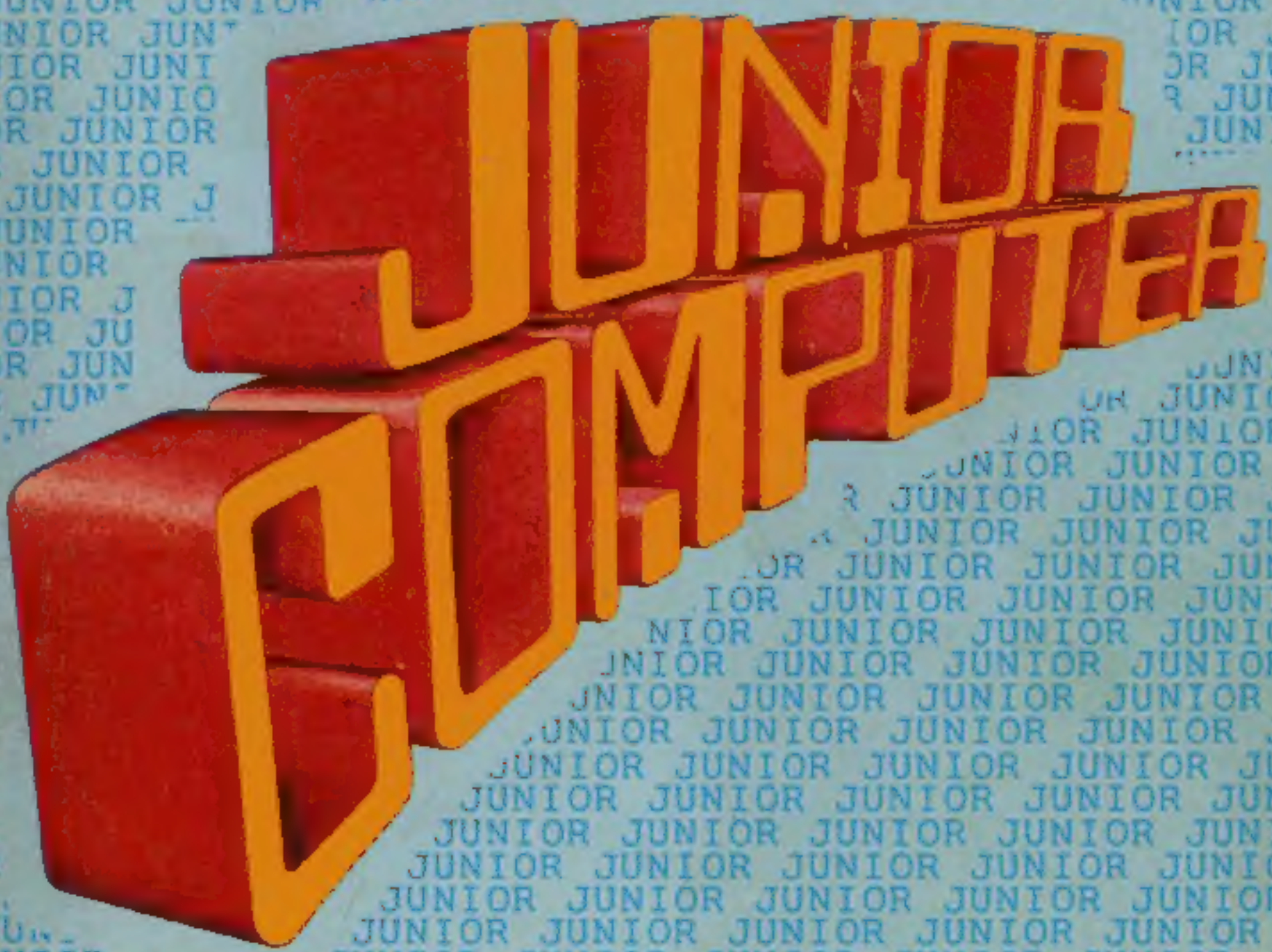
Der einfache Einstieg in die faszinierende Computertechnik



Buch 3

Elektor Verlag

28,50

Junior-Computer 3

Der einfache Einstieg in die faszinierende Computertechnik

A. Nachtmann
G.H. Nachbar

ISBN 3-921608-24-4

Elektor Verlag GmbH, 5133 Gangelt 1

Über die Liefermöglichkeit der im Buch veröffentlichten Platinen gibt die jeweils gültige EPS-Liste (Elektor-Platinen-Service) in der Zeitschrift ELEKTOR Auskunft. Das gilt auch für weitere Informationen über den Elektor-Software-Service (ESS).

© 1981 Elektor Verlag GmbH, 5133 Gangelt 1

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch teilweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet. Alleiniges Nachdruckrecht für das holländische Sprachgebiet: Elektuur B.V., Beek (L), Holland; für Publikationen in englischer Sprache: Elektor Publishers Ltd., Canterbury, England; für das französische Sprachgebiet: Elektor sarl, Estaires, Frankreich. Der Nachbau der veröffentlichten Schaltungen geschieht außerhalb der Verantwortlichkeit des Herausgebers.

Gezieltes Wachstum!

Das Motto dieses nun vor Ihnen liegenden dritten Buchs lautet: "Vom Junior-Junior-Computer zum Senior-Junior-Computer" oder "Vom Einplatinen-Mikrocomputer für Anfänger zum kompletten Mikrocomputersystem für Fortgeschrittene".
Wir meinen damit folgendes: Für den einen ist der Ausbau des Junior-Computers an Hand der Bücher 1 und 2 abgeschlossen, für den anderen hat er gerade erst begonnen. Der Appetit stellt sich oft erst beim Essen ein, so daß die Mehrzahl der Junior-Computer-Freunde sicherlich der zweiten Gruppe angehört. Hier findet sich noch einmal reichlich Nahrung und, anders als in Buch 2, ist auch der Lötkolben wieder mit im Spiel.
Wachstum in vernünftigen Grenzen, das hatten wir uns zum Ziel gesetzt. Ein Ausbau des Juniors zum Großrechenzentrum war nicht geplant. Das Ergebnis ist eine ausgewogene und aufeinander abgestimmte Mischung von zusätzlicher Hardware und Systemsoftware, von neuen Lötstellen und neuen Bytes.
Und hier die Erweiterungen im einzelnen:
Punkt 1. Das wiederholte Eintippen eines häufig benutzten Programms ist mit dem mehrfachen Schreiben des selben Buchs vergleichbar, bevor es erneut gelesen werden kann. Ein Buch muß aber nur ein einziges Mal geschrieben werden, – oder etwa nicht? Als Notizbuch für Computerprogramme und andere Daten ist die Magnetkassette hervorragend geeignet. Das Schreiben in dieses Notizbuch und das beliebig häufige Lesen des Geschriebenen erledigt der Computer mit Hilfe der Kassetteninterface-Hardware, zusammen mit der Software des Programms Tape Management.
Punkt 2. Geringe Merkfähigkeit des Gedächtnisses hat beim Menschen andere Ursachen als beim Computer. Die "grauen Zellen" des Computers lassen sich bis zu einer bestimmten oberen Grenze vermehren. Das ist notwendig, wenn längere Programme laufen sollen oder mehr Computer-Intelligenz erwünscht ist. Beides wird durch zusätzliche Speicher-ICs (RAMs bzw. EPROMs) realisiert.
Punkt 3. Das hexadezimale Tastenfeld mit seinen 23 Funktionen und das sechstellige Siebensegmentdisplay können auf Wunsch durch eine ASCII-(Schreibmaschinen-)Tastatur und ein Datensichtgerät (Videodisplay) ergänzt werden. An das dafür notwendige RS-232-Interface ist auch der Anschluß eines Druckers und anderer peripherer Geräte möglich. Nicht nur der Bedienungskomfort steigt dadurch beträchtlich an, sondern gleichzeitig ist eine der Voraussetzungen geschaffen, die für das Arbeiten mit höheren Programmiersprachen (z.B. BASIC) erfüllt sein müssen.

Punkt 4. Der bei der Junior-Basisversion bereits vorhandene PIA wird nur noch für computerinterne Aufgaben verwendet. Die vielfältigen Verbindungen zur Außenwelt stellt nun ein weiterer I/O-Baustein, ein VIA her. Dieser "Versatile Interface Adapter" ist noch vielseitiger als der PIA; er steht voll und ganz im Dienst des Programmierers.
Das Buch ist in drei Kapitel aufgegliedert: In Kapitel 10 beschäftigen wir uns hauptsächlich mit der Arbeitsweise und dem Bau der Erweiterungen, also mit der Hardware. Anschließend lernen wir in Kapitel 11, wie man die Bandkassette als Datenspeicher einsetzt und welche Rolle das Systemprogramm Tape Management dabei spielt. Während bis zu diesem Punkt nur ein oder zwei Kassettenrekorder extern an den Junior angeschlossen sind, wächst in Kapitel 12 die Peripherie sprunghaft: Es wird der Anschluß des Videoterminals und des Druckers beschrieben. Dazu gehört der Umgang mit dem Systemprogramm Printer Monitor. Beispiele aus der Praxis, genaue Anleitungen zum Ausprobieren und Demonstrieren sowie Ratschläge für eine eventuelle Fehlersuche sind in alle Kapitel eingestreut, so daß sich "Wachstumshemnisse" schnell beseitigen lassen.
Der Junior-Computer war bisher ein Einplatinen-Mikrocomputer, maßgeschneidert für den Einstieg in das Computer-Hobby. Er reift in diesem Buch zum vollwertigen System heran, das seinen Kinderschuhen längst entwachsen ist.

Die Autoren

P.S. Abgesehen von einigen "Hard Copies" wurde dieses Buch nicht von einem Computer, sondern von Menschen geschrieben. Die Autoren sind für schriftliche Hinweise, für Kritik und natürlich auch für Lob dankbar.

Inhalt

Kapitel 10 7
Der Junior wächst – *nach alles auf einer Karte nun noch mehr auf mehreren Karten.*

Kapitel 11 80
Magnetische Speichererweiterung – *Daten vom, zum und am laufenden Band*

Kapitel 12 120
Junior auf dem Bildschirm – *Mehr Sicht- und Tastbares*

Anhang 1 182
Basiskarte und RAM/EPROM-Karte – *Speichererweiterung ohne Interface-karte*

Anhang 2 186
PIA-RAM-Erweiterung für den Standard-Monitor – *Schritt für Schritt dezimal rechnen*

Anhang 3 191
Vektoren und Buskartenspeicher – *Alternativen ohne EPROM*

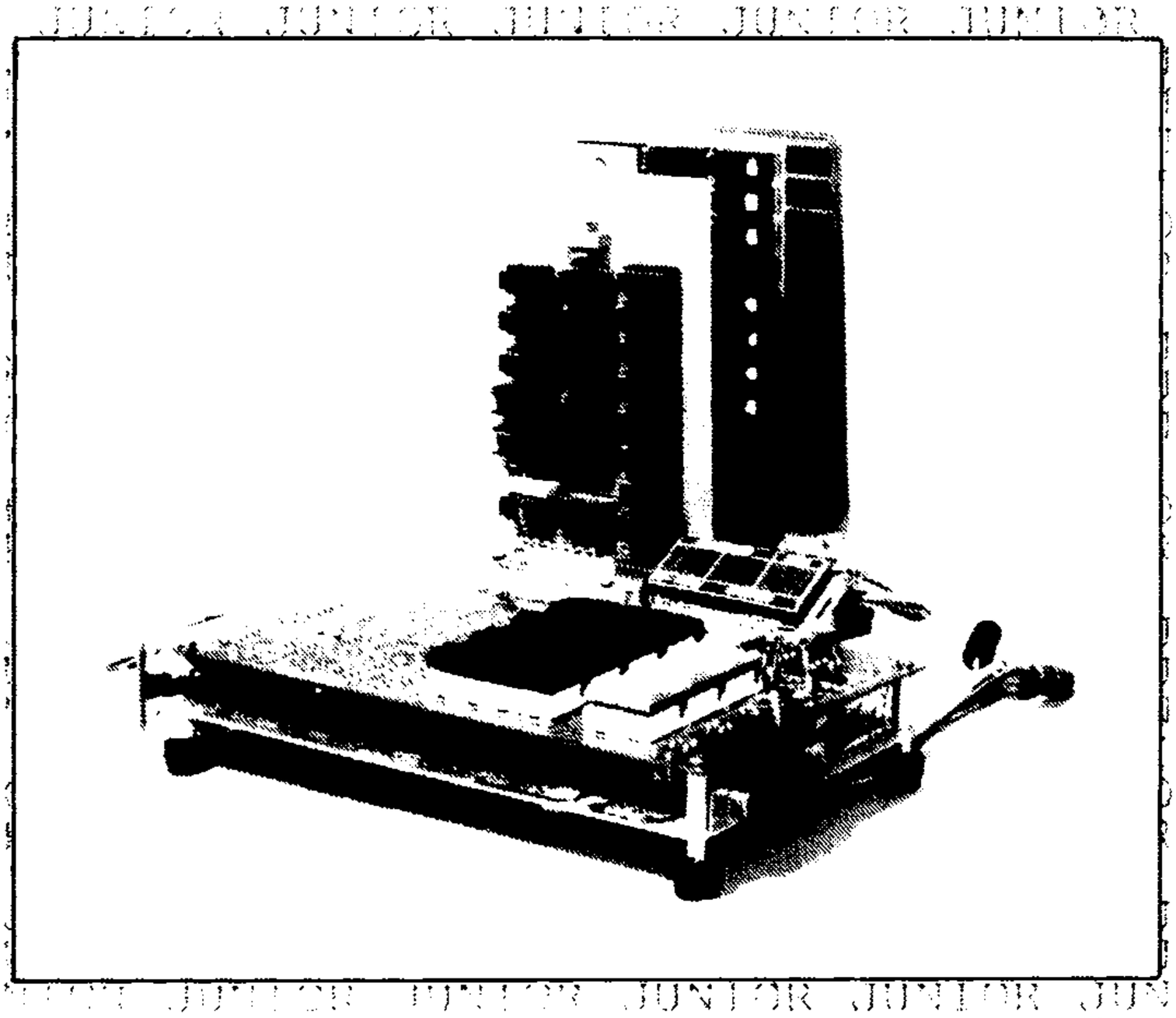
Anhang 4 195
Hexdump vom Programm Tape Monitor (TM)

Anhang 5 197
Hexdump vom Programm Printer Monitor (PM)

Anhang 6 199
RAM-Testprogramm – *Schreib- und Lesetest*

Anhang 7 206
ASCII-Codetabelle

Sachwortverzeichnis 207



10

Der Junior wächst

nach
alles auf einer Karte
nun
noch mehr auf mehreren Karten

Der Junior-Computer wächst nicht kontinuierlich, sondern karten(platinen-)weise. In diesem Kapitel geht es um die einzelnen Karten sowie um die Regeln des Kartenspiels. Wenn alle Karten auf dem Tisch liegen, hat der Junior-Computer folgende Eigenschaften hinzugewonnen:

- eine Menge digitaler "grauer Zellen" (RAM, EPROM bis maximal 64 KByte);
- vielseitige Verbindungen zur Außenwelt durch einen zweiten I/O-Baustein (VIA) und einen seriellen Datenein- und -ausgang (RS-232-Interface), an den ein Videoterminal mit ASCII-Tastatur oder ein Drucker angeschlossen werden kann.

Die Interfacekarte ist die Trumpfkarte unter den Kartenblättern, ihr fällt eine Schlüsselrolle zu. Der erste Teil dieses Kapitels erklärt, welche Funktionen die Interfacekarte hat und wie sie diese erfüllt. Im zweiten Teil folgt dann die ausführliche Bauanleitung. Die übrigen am Spiel beteiligten Karten sind mit bereits früher in Elektor beschriebenen Karten verwandt. Dazu gehört die RAM/EPROM-Speicherkarte; sie wird hier ebenfalls besprochen. Übrigens müssen nicht sämtliche Erweiterungskarten sofort vorhanden sein: ein allmählicher Ausbau des Systems schont den Geldbeutel. Vorläufige oder sogar endgültige Abstriche sind vor allem bei der Bemessung der Speicher-

kapazität möglich. Auch das kommt in diesem Kapitel zur Sprache.
Der Junior-Computer wächst. Aus dem Junior wird nun ein ausgereifter Senior!

Wachstum ist ein natürlicher Vorgang, für den die Natur im Bereich der Lebewesen selbst sorgt. Dagegen ist die Triebfeder für das Wachsen des Junior-Computers meistens in der Natur seines Besitzers begründet. Elektor gibt dazu das Know-How, so daß nichts schiefgehen kann. Bereits in Buch 1 wurde der Computer mit dem Menschen verglichen: Die Hardware ist des Menschen Fleisch und Blut, die Software ist sein Geist. Nach Erreichen eines bestimmten Lebensalters kommt bekanntlich das körperliche Wachstum des Menschen zum Stillstand, das geistige Wachstum kennt dagegen keine Altersgrenze. Übertragen auf den Junior-Computer bedeutet dies, daß der Junior nun die Schwelle zum Erwachsenenalter überschreiten kann. Mit der in diesem Buch beschriebenen Hardware ist sein körperliches Wachstum im wesentlichen abgeschlossen.
Wenn der Junior-Computer dieses Ausbaustadium erreicht hat, können an ihn unter anderem ein Kassettenrekorder, eine ASCII-Tastatur, ein Video-Interface und ein Drucker angeschlossen werden. Das äußere Wachsen macht auch ein inneres Wachsen, die Erweiterung der Systemsoftware, notwendig. Beides zusammen eröffnet neue Dimensionen für das weitere geistige Wachstum: Beispielsweise kann der Junior-Computer nun "fremde" Sprachen lernen. Obwohl der Computer weiterhin in seiner Muttersprache (der Maschinensprache) "denkt", drückt er sich auf dem Papier des Druckers oder auf dem Bildschirm in einer höheren Sprache, zum Beispiel in BASIC aus. Wesentliche Änderungen oder Ergänzungen der Hardware sind dafür nicht mehr erforderlich; das Wachsen der Intelligenz läßt seine äußere Gestalt unverändert.

Bus, Karte und Buskarte

Es war bereits die Rede davon, daß Wachsen ein natürlicher Vorgang ist. Während dies in der Natur von selbst geschieht, wächst der Junior-Computer nur durch die Tatkraft seines Besitzers. Er kann beschließen, den Computer in seiner Grundausbaustufe zu belassen, ihn allmählich Schritt für Schritt zu ergänzen oder sofort das komplette System mit allem Zubehör aufzubauen. Dabei stellt sich häufig die Frage, bis zu welchem Stadium ein Ausbau überhaupt sinnvoll und notwendig ist. Diese Frage kann konkret nur im Einzelfall beantwortet werden; es kommt hier ganz erheblich auf den Einsatzzweck des Computers und die Ansprüche seines Besitzers an.

Eine oder mehrere Karten?

Hobbycomputer für den Selbstbau haben entweder die Gestalt eines Bussystems oder eines Einkartencomputers. Ein Bussystem besteht aus mehreren Karten (Platinen), die meistens die gleichen äußeren Abmessungen besitzen. Verbreitet ist auch hier das Format 16 x 10 cm, das sogenannte

Eurokartenformat. Die einzelnen Karten sind miteinander über ein Leitungsbündel (den Bus) in der Weise verbunden, daß jeder Anschluß einer Karte mit dem entsprechenden Anschluß auf den anderen Karten in Verbindung steht. In der Praxis werden dafür entweder einzelne Drähte oder aber eine weitere Platine, die Buskarte, verwendet. Da die Länge des Bus **im Prinzip** nicht begrenzt ist, kann das System beliebig ausgebaut werden. Auf diese Weise entsteht ein Computersystem mit fast unbegrenzten Einsatzmöglichkeiten.
Bei **Einplatinensystemen** (Single-Board-Systemen) wurde vom Entwickler buchstäblich alles auf eine Karte gesetzt. Hier entscheidet hauptsächlich die Bestückung und damit auch das äußere Format der Platine, welche Möglichkeiten der Computer letztendlich bietet und wo seine Grenzen liegen. Diese Grenzen lassen sich mit vertretbarem Aufwand meistens nicht weiter hinausschieben.
Einkartencomputer erfüllen ihre Aufgabe als Übungs- und Lehrcomputer sowie bei der Steuerung von einfachen Prozessen. Der Junior-Computer wurde so konzipiert, daß er in seiner Basisversion ein Einkartencomputer mit den genannten Möglichkeiten ist. Zahllose Leser der Junior-Computer-Buchreihe lassen ihn bereits in dieser Gestalt arbeiten. Ihr Interesse am Computerhobby wurde häufig durch die Veröffentlichungen und Beiträge in den Medien geweckt, die sich mit Chips, Mikros und ähnlichen Wundern moderner Technik beschäftigen. Nicht selten bleibt es aber bei einfachen oder einseitigen Anwendungen des zusammengebauten Juniors: beim einen dient er als A/D-Wandler; im anderen Fall, auch das wurde bekannt, leistet er Hilfe bei der Steuerung von Prozessen in der Halbleiterherstellung. Natürlich ist generell nichts dagegen einzuwenden, daß der Junior-Computer in seiner Grundausbaustufe bleibt und eine stets unveränderte Aufgabe erfüllt. Auch als Einkartencomputer bietet der Junior unzählige Möglichkeiten; sein Erweiterungskonnektor kann deshalb leicht in Vergessenheit geraten.

Die wichtigsten Erweiterungen

Die Kontaktleiste an der Schmalseite der Hauptplatine schafft die Voraussetzung dafür, daß der Junior zum Mehrplatinencomputer ausgebaut werden kann. Dieser Konnektor ist die Schnittstelle für folgende Erweiterungen:

- **Mehr Speicherkapazität.** Zusätzliche RAMs sind notwendig, um längere Anwenderprogramme verarbeiten zu können. Auch das Laden umfangreicher System- und Übersetzerprogramme (Interpreter) vom Band erfordert einen Ausbau der RAM-Kapazität. Residente Systemprogramme müssen dagegen in weiteren EPROMs untergebracht werden. Dazu zählen die Systemprogramme TM (Kapitel 11) und PM (Kapitel 12).
- **Mehr I/O-Leitungen.** Der PIA auf der Hauptplatine erhält einen Kollegen, so daß der Datenaustausch mit der Außenwelt auf breiteren Bahnen abgewickelt werden kann.
- Hardware für die Nutzung von Magnetbändern als **Datenarchiv** (Hintergrundspeicher) und als Programmbibliothek. Ein Archiv enthält lauffähige oder noch unvollendete Software-Bausteine aus eigener Produktion, während mit **Programmbibliothek** meistens eine Sammlung von gekauften, geliehenen oder kopierten Programmen gemeint ist. Das können sowohl

Anwender- als auch Systemprogramme sein (letztere müssen nicht unbedingt in einem EPROM stecken); ein BASIC-Interpreter läßt sich auch vom Band laden, wenn er nicht ständig benötigt wird. Die Hardware, die den Datenfluß vom Band zum RAM und umgekehrt ermöglicht, wird auch Kassetteninterface genannt. Um Software vom Band laden zu können, ist eine entsprechend bemessene RAM-Speicherkapazität zwingend notwendig. Die Kassetteninterface-Hardware wäre ohne Mithilfe der Kassettensoftware nicht arbeitsfähig. Beim Junior-Computer ist dies das Systemprogramm TAPE MONITOR oder TAPE MANAGEMENT (TM), das in Kapitel 11 besprochen wird.

- Hardware für die Kommunikation mit peripheren Geräten, die nach einer bestimmten Norm Daten senden und empfangen: **das RS-232-Interface.** Eine ASCII-Tastatur bietet mehr Möglichkeiten und mehr Bequemlichkeit bei der Bedienung des Computers als das hexadezimale Tastenfeld auf der Basisplatine. An die Stelle der sechs Siebensegmentdisplays für die Datenausgabe können ein Drucker oder der Bildschirm eines Terminals treten. Hierfür ist insbesondere das von Elektor publizierte "Elekterminal" geeignet. Für eine solche Ein- und Ausgabeeinheit werden ebenfalls neue Systemprogramme wie Texteditor, Assembler, Disassembler usw. benötigt. Das Systemprogramm PRINTER MONITOR (PM), beschrieben in Kapitel 12, enthält sämtliche dafür notwendige Software. Die Voraussetzungen für das Arbeiten mit BASIC sind damit, soweit sie die Hardware betreffen, ebenfalls geschaffen.

Nach Anschluß einer weiteren Platine gehört der Junior-Computer streng betrachtet nicht mehr zur Gruppe der Einkartencomputer. Das bedeutet jedoch keinesfalls, daß der Junior nun die Gestalt eines "endlosen" Bussystems annehmen soll. Die Hardware-Erweiterungen haben lediglich abrundenden Charakter; sie steigern die Leistungsfähigkeit und erhöhen den Bedienungskomfort auf ein Maß, das für den universellen Einsatz des Juniors notwendig ist. Die später noch zu beschreibende 16-K-Speicherkarte mit dynamischen RAMs erfüllt keine neue Funktion, sondern stellt nur eine preiswerte Alternative für zwei 8-K-RAM-EPROM-Karten dar.
Der Junior-Computer ist somit kein Bussystem im üblichen Sinn. Abgesehen von den peripheren Einheiten (Bandgerät, Drucker, Terminal usw.) und einem Teil der zusätzlichen Speicher-ICs (sofern vorhanden) befinden sich alle Hardware-Erweiterungen auf einer einzigen Platine: der Interfacekarte. Diese Platine hat die gleichen Abmessungen wie die Basisplatine des Junior-Computers, so daß beide Karten miteinander verschraubt eine Art "Sandwich-Konstruktion" bilden können.
Auf die Interfacekarte kann bei Bedarf eine Busplatine gesteckt werden, die als Träger für weitere Speicherkarten dient. Der Doppelplatinen-Computer wird damit zum Bussystem. Letzten Endes hängt es deshalb von den individuellen Bedürfnissen des Besitzers ab, ob der Junior zum Doppelkarten- oder zum Bussystem heranwächst. Wichtiger als die Zugehörigkeit zu einer bestimmten Computerfamilie ist jedoch die Beschreibung dessen, was den Einkartencomputer zum Doppelkartencomputer macht und gleichzeitig die Voraussetzung für den Ausbau zum Bussystem schafft.

Die Interfacekarte

Interface bedeutet so viel wie Kopplung, Anpassung oder Bindeglied. Die Interfacekarte des Junior-Computers stellt die Verbindungen zu zwei Kassettenrekordern und zu weiteren peripheren Geräten entsprechend der Norm RS-232 her; sie enthält ferner einen Datenbuspuffer für die Basiskarte, der als internes Interface betrachtet werden kann. Die außerdem noch vorhandenen Speicherbausteine ergänzen die Speicherkapazität der Hauptplatine. Die Bilder 1 und 2 zeigen die Schaltung der Interfacekarte. Wir beginnen mit der Besprechung von Bild 1.

Stoßstellen: die Datenbuspuffer

In Bild 1 sind links die Anschlüsse des Eingangskonnektors (INPUT CONNECTOR) gezeichnet. Über die Stifte dieser Kontaktleiste sowie über fünf weitere Verbindungen mit dem PORT CONNECTOR (siehe Bild 2) verläuft die Kommunikation mit der Basisplatine des Junior-Computers. Bis auf die Leitungen EX und K1 . . . K6 (diese enden auf der Interfacekarte) finden sich sämtliche Leitungen auf dem Ausgangskonnektor (OUTPUT) CONNECTOR) wieder, der in Bild 1 rechts oben dargestellt ist. Der Ausgangskonnektor verbindet die Interfacekarte mit der Buskarte, die ihrerseits die Verbindungen zu einer oder mehreren Speicherkarten herstellt. Ein Signal, das an einem Stift mit der Bezeichnung a des Eingangskonnektors liegt, erscheint am Ausgangskonnektor an einem Stift mit der Bezeichnung c, und umgekehrt. Warum dies so ist, wird in der Bauanleitung für die Interfacekarte erläutert.
Die Adreßleitungen AØ . . . A15 und die Datenleitungen DØ . . . D7 des Eingangskonnektors erscheinen auf dem Ausgangskonnektor als ABØ . . . AB15 und DBØ . . . DB7. Der Buchstabe B steht für Buffer (Puffer); die Adreß- und Datenleitungen werden nämlich auf der Interfacekarte gepuffert.
Pufferung, was ist das eigentlich? Die Antwort ist einfach: Wie bei einem Eisenbahnwaggon fangen die Puffer auch hier Belastungsstöße auf; sie sorgen für weitgehende Unabhängigkeit der Signalspannungen von der Größe der angeschlossenen Last. Fehlt eine ausreichende Pufferung, so werden auch die Umschaltzeiten länger. Der Computer verhält sich dann wie ein Lastwagen, in den der Motor eines Mofa eingebaut wurde: die Beschleunigung beim Anfahren ist nicht gerade überragend. Ohne Pufferung werden die Daten- und Adreßleitungen des Junior-Computers überlastet. Die Spannungen, die die logischen Signale Ø und 1 ausdrücken, werden bei Signaländerungen relativ langsam oder überhaupt nicht erreicht. Zeit ist aber bei einem Computer (man denke an den 1-MHz-Takt; ein "Herzschlag" pro millionstel Sekunde) eine außerordentlich wichtige Größe. Eine Verzögerung von zehn Nanosekunden (zehn milliardstel Sekunden!) kann den Datenfahrplan bereits durcheinanderbringen.
Ein zweiter Grund für die Pufferung ist folgender: Unter bestimmten Voraussetzungen muß der Datenfluß auf den Datenleitungen in einer oder auch in beiden Richtungen blockiert werden. Die Daten werden in Richtung zum Mikroprozessor gelesen und in umgekehrter Richtung geschrieben, während die Adressierung stets nur vom Mikroprozessor ausgeht

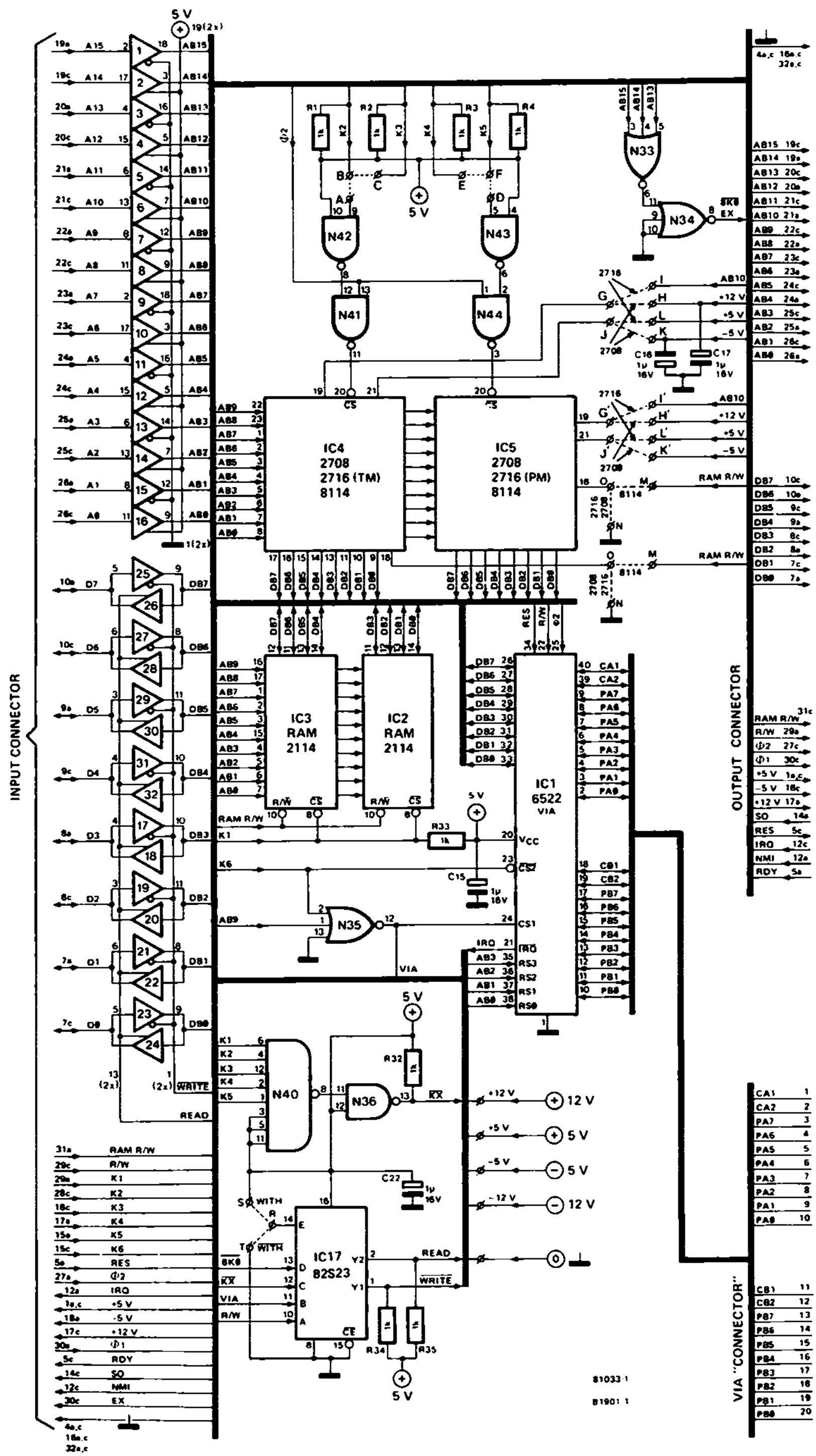


Bild 1. Auf der Interfacekarte sind unter anderem der Datenbuspuffer, zusätzliche Speicher-ICs und ein weiterer I/O-Baustein untergebracht. Das Bild zeigt die Schaltung der genannten Funktionseinheiten.

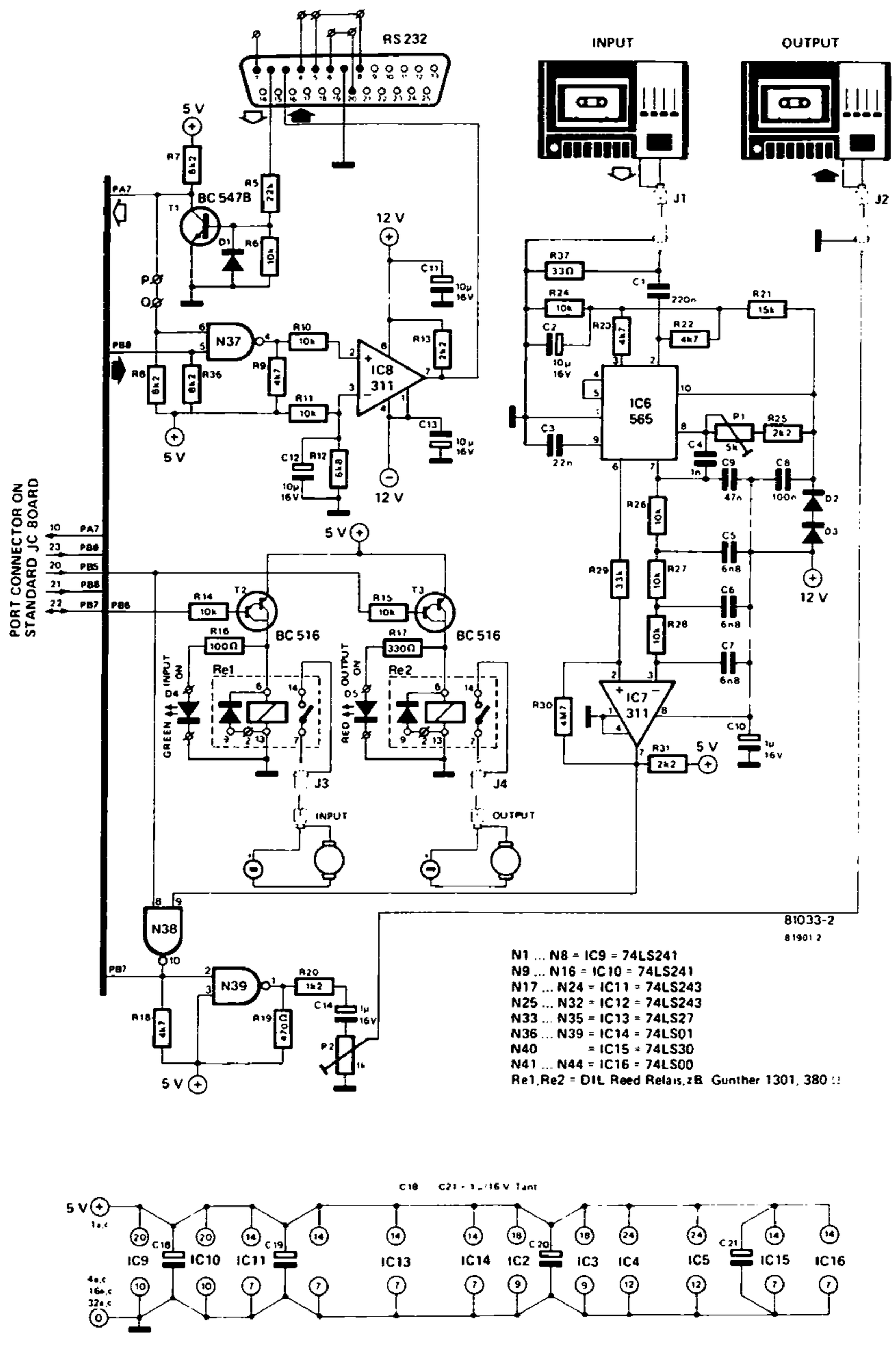


Bild 2. Auch diese Schaltung befindet sich auf der Interfacekarte: das Kassetten-interface für zwei Rekorder und das RS-232-Interface zum Anschluß peripherer Geräte.

und deshalb nur in einer Richtung verläuft.
Das Symbol für einen Puffer ist ein Dreieck, das ähnlich einer Pfeilspitze auf den Ausgang zeigt; der gegenüberliegende Anschluß ist der Eingang. In Bild 1 sind 32 Puffer gezeichnet: die Adreßpuffer N1 . . . N16 und die Datenpuffer N17 . . . N32. Da die Adressierung stets vom Prozessor ausgeht, befindet sich in Bild 1 der Eingang von N1 . . . N16 jeweils links und der Ausgang jeweils rechts. Von den Datenpuffern sind dagegen pro Datenleitung zwei Puffer parallel, jedoch in entgegengesetzter Richtung geschaltet. Die Ausgänge der Puffer mit geradzahliger Numerierung weisen nach links, die Ausgänge der Puffer mit ungeradzahliger Numerierung weisen nach rechts. Wie schon angedeutet, haben die Datenpuffer zusätzlich die Funktion von Verkehrspolizisten. Bild 3 verdeutlicht, was damit gemeint ist. Wir setzen den Verkehrsfluß mit einem elektrischen Strom und die Steuerung durch den Polizisten mit der Funktion von zwei Schaltern gleich, die in Serie mit einer Diode liegen. Den Mikroprozessor denken wir

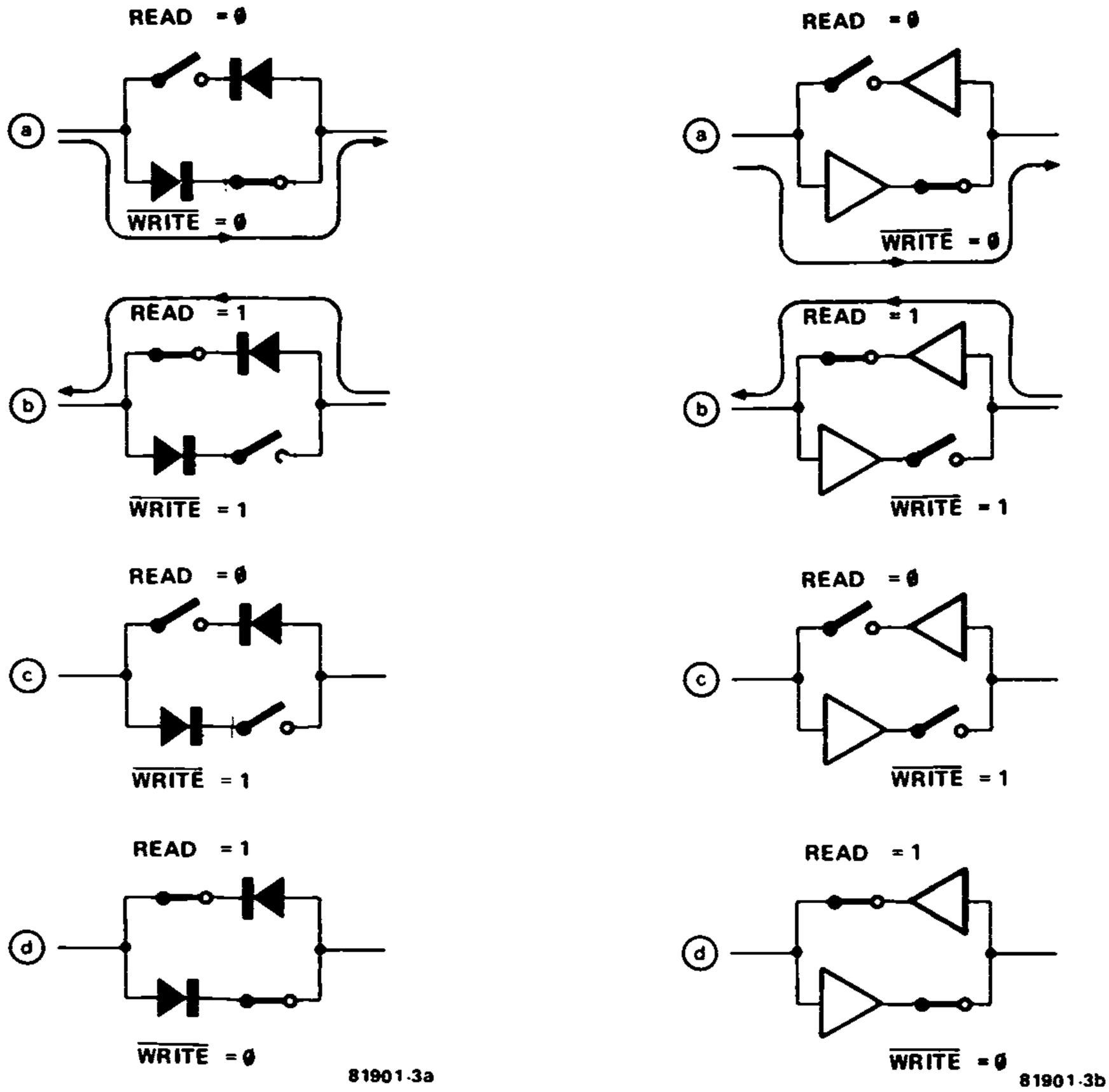


Bild 3. Vergleich des Datenbuspuffers mit zwei antiparallelgeschalteten Dioden und zwei Schaltern. Der hindurchfließende Strom stellt den Datenfluß dar, während die Schalter die Wirkung der Steuersignale nachbilden.

uns links von dieser Anordnung, während sich die Erweiterungen rechts befinden. Die Schalter verkörpern die Signale $\overline{WRITE}$ und READ, wobei ein offener Schalterkontakt der logischen Ø und ein geschlossener Schalterkontakt der logischen 1 entspricht. Insgesamt sind vier Zustände möglich, von denen jedoch nur drei sinnvoll sind:

1. Die Puffer mit ungerader Numerierung N17, N19 usw. . . . N31) sind aktiv, während die übrigen Puffer inaktiv sind. Dieser Fall tritt dann ein, wenn die Signale $\overline{WRITE}$ und READ auf logisch Ø gehen. Der Datenfluß ist jetzt von der Basiskarte nach außen zu den übrigen Funktionseinheiten des Computers gerichtet. Dieser Zustand entspricht der Schalterstellung ⓐ in Bild 3.

2. Die Puffer mit ungerader Numerierung sind inaktiv (Tri-State-Zustand; die Ausgänge sind hochohmig), während die Puffer mit gerader Numerierung (N18, N20 usw. . . . N32) die aktive Rolle übernehmen. Dazu müssen die Leitungen $\overline{WRITE}$ und READ beide logisch 1 sein. In diesem Zustand können Daten von außen (z.B. von der Interfacekarte) zur Basiskarte fließen. Entspricht der Schalterstellung ⓑ in Bild 3.

3. Sämtliche Puffer sind inaktiv, wenn das Signal $\overline{WRITE}$ logisch 1 und das Signal READ logisch Ø ist. Ein Austausch von Daten mit der Basisplatine kann nicht stattfinden. Dieser Zustand ist immer dann notwendig, wenn Daten nur intern zwischen den auf der Basisplatine selbst befindlichen Bausteinen zu transportieren sind. Entspricht Schalterstellung © in Bild 3.

4. Die vierte Möglichkeit, bei der sämtliche Datenpuffer gleichzeitig aktiv sind ($\overline{WRITE}$ = Ø und READ = 1), ist nicht sinnvoll, da nicht gleichzeitig gelesen und geschrieben werden kann. Von ihr wird deshalb kein Gebrauch gemacht. Entspricht Schalterstellung ⓓ in Bild 3.

Zweiter I/O-Baustein: der VIA

Das Innenleben von IC1 in Bild 1, einem VIA (Versatile Interface Adapter) vom Typ 6522, ist ein Kapitel für sich; ihm wurde deshalb eine detaillierte Beschreibung in Junior-Computer-Buch 4 gewidmet. Hier soll nur angemerkt werden, daß dieser Baustein den PIA (Peripheral Interface Adapter) des Typs 6532 auf der Basisplatine an Vielseitigkeit noch erheblich übertrifft.

Die wichtigen Verbindungen mit der Außenwelt sind auf den VIA-"CONNECTOR" gelegt (siehe Bild 1). Da dieser Konnektor kein echter Konnektor ist (er befindet sich nicht auf der Platine), wurde "CONNECTOR" in Anführungszeichen gesetzt. Der 6522 ist mit den Adreßleitungen ABØ . . . AB3 und mit verschiedenen Leitungen des Steuerbus verbunden. Natürlich sind am 6522 auch die acht Leitungen des Datenbus sowie die IRQ-Leitung angeschlossen. Über letztere wird ein Interrupt ausgelöst, wenn zum Beispiel ein im 6522 integrierter Timer den "time out" erreicht.

Ebenso wie der 6532 hat auch der 6522 zwei Chip-Select-Signale. Die Eingänge $\overline{CS2}$ beider Bausteine liegen gemeinsam an Ausgang K6 des Adressendekoders IC6, der sich auf der Basisplatine befindet. Der Eingang CS1 des 6522 ist dagegen mit dem Ausgang von NOR-Gatter N35 verbunden. Den Eingängen von N35 werden die Signale K6 und AB9 zugeführt. Der 6522 ist nur dann aktiv, wenn $\overline{CS2}$ logisch Ø und gleichzeitig CS1

logisch 1 ist. Letzteres bedeutet, daß AB9 und K6 logisch Ø sein müssen (der 6532 ist dagegen bei A9 = 1 aktiv). Berücksichtigt man ferner, daß Leitung A8 (bzw. AB8) weder mit dem 6522 noch mit dem 6532 verbunden ist, dann ergibt sich für den VIA (6522) auf der Interfacekarte folgender Adressenbereich:
18ØØ . . . 19FF
(AB8 = X; AB9 = Ø; K6 = Ø)
Der PIA (6532) auf der Basisplatine hat dagegen den Adressenbereich
1AØØ . . . 1BFF
(A8 = X; A9 = 1; K6 = Ø).
Läßt man die Doppeladressierung durch A8 = X außer Betracht, sind sowohl für den PIA als auch für denVIA jeweils 256 Adressen verfügbar. Doch ebenso wie beim PIA (6532), der 19 verschiedene, mehrfach adressierbare "Speicherplätze" besitzt, sind auch beim VIA nicht sämtliche 256 Adressen belegt. Wie aus Bild 1 hervorgeht, stehen die Adreßleitungen AB4 . . . AB7 mit dem VIA nicht in Verbindung; beim VIA können somit nur 16 Adressen angesteuert werden.
Warum haben wir einen zweiten I/O-Baustein hinzugefügt? Hier lautet die Antwort wie folgt: Der PIA ist hauptsächlich mit internen Vorgängen wie der Steuerung der sechs Siebensegmentdisplays und der Abfrage des hexadezimalen Tastenfelds beschäftigt. Hinzu kommen neue Aufgaben, die in Zusammenhang mit dem Kassetten- und dem RS-232-Interface stehen. Dadurch wird die Verwendbarkeit des PIA innerhalb von Programmen eingeschränkt, die den Monitor mitbenutzen. Will man beispielsweise ein Programm, wie es in Kapitel 6 von Buch 2 angegeben ist, Schritt für Schritt auf dem Computer laufen lassen, dann können leicht Probleme auftreten. Für kritische Anwendungen des Timers gilt dies natürlich erst recht. Beim Step-by-Step-Mode folgt nämlich auf jeden Befehl ein Sprung zum SAVE-Eingang des Monitors. In Kapitel 7 von Buch 2 ist dies ausführlich beschrieben. Wenn der Zustand der Ports vollständig oder auch nur teilweise von dem Zustand abweicht, den die richtige Arbeitsweise des Monitors erfordert, kann der Monitor nicht einwandfrei laufen.
Ähnliche Komplikationen können sich auch in verschiedenen anderen Fällen ergeben. Sie lassen sich vermeiden, wenn der PIA für interne Aufgaben reserviert bleibt (die Monitor-Subroutinen sind natürlich trotzdem für Anwenderprogramme nutzbar); der PIA übernimmt dann den "Außendienst".

Kurzes Intermezzo: Adreßdekodierung

Adressen und Postleitzahlen

So wie die Wohnungsadressen jeder Stadt einen gemeinsamen Bestandteil, nämlich die Postleitzahl haben, tragen auch bestimmte Gruppen von Adressen im Junior-Computer ein gemeinsames Kennzeichen. Dieses Kennzeichen ist das Selektionssignal. Die auf der Basisplatine und auf der Interfacekarte befindlichen Bausteine werden von den Signalen KØ . . . K7 selektiert. Geliefert werden diese Signale von Adreßdekoder IC6 auf der Basisplatine. Auch die auf die Interfacekarte aufgesteckten Speicherkarten benötigen Selektionssignale; sie werden jedoch auf den Speicherkarten selbst erzeugt (siehe Besprechung der RAM/EPROM-Karte am Schluß

dieses Kapitels).
Rufen wir uns noch einmal die Arbeitsweise von Adressendekoder IC6 ins Gedächtnis zurück. In Bild 4 ist er als mechanischer Vielfachumschalter dargestellt, während Tabelle 1 die dazugehörige Wahrheitstabelle wiedergibt. Wenn eins der acht Selektionssignale KØ . . . K7 durch Anlegen der entsprechenden Eingangssignale aktiviert wird, ist dieses Selektionssignal logisch Ø; die übrigen sieben K-Ausgänge werden dagegen durch Widerstände auf logisch 1 gezogen. Die logische Ø ist hier deshalb das aktivierende Signal, weil die meisten Bausteine durch eine Ø an ihren $\overline{CS}$-Eingängen aktiviert werden.
Der Schalter in Bild 4 besitzt 16 Kontakte, von denen jedoch nur acht angeschlossen sind; insgesamt sind 16 Schalterstellungen möglich. Welche Stellung der Schalter einnimmt, hängt von den Steuersignalen A, B, C und D ab. Während A, B und C mit den Leitungen A10, A11 und A12 des Adreßbus verbunden sind, liegt Eingang D entweder an Masse (logisch Ø) oder Anschluß EX des Erweiterungskonnektors. Bestimmend ist hierfür die Lage einer Drahtbrücke auf der Basisplatine.
Liegt Steuereingang D an Masse, was einer logischen Ø entspricht, dann kann sich der Arm des Schalters in Bild 4 über die Kontakte Ø . . . 7 bewegen. Ist D mit EX verbunden, so bestimmt das Signal EX, ob ein Kontakt aus der Gruppe Ø . . . 7 (EX = logisch Ø) oder aus der Gruppe 8 . . . 15 (EX = logisch 1) vom Schalter an Masse gelegt wird. Aus der rechten Spalte der Wahrheitstabelle (Tabelle 1) geht hervor, daß nur im Fall D = Ø ein Baustein auf der Basisplatine oder auf der Interfacekarte aktiviert wird. Bei D = 1 werden dagegen die Speicherbausteine angesprochen, die nicht an IC6 angeschlossen sind und auf der Buskarte stecken. Wenn solche zusätzlichen Karten vorhanden sind, muß D mit EX verbunden werden; fehlen sie, so muß die Drahtbrücke ihre ursprüngliche Lage behalten. Dies hat seinen Grund darin, daß sämtliche auf der Basiskarte und auf der Interfacekarte vorhandenen Bausteine über die acht Selektionssignale KØ . . . K7 adressiert werden können.

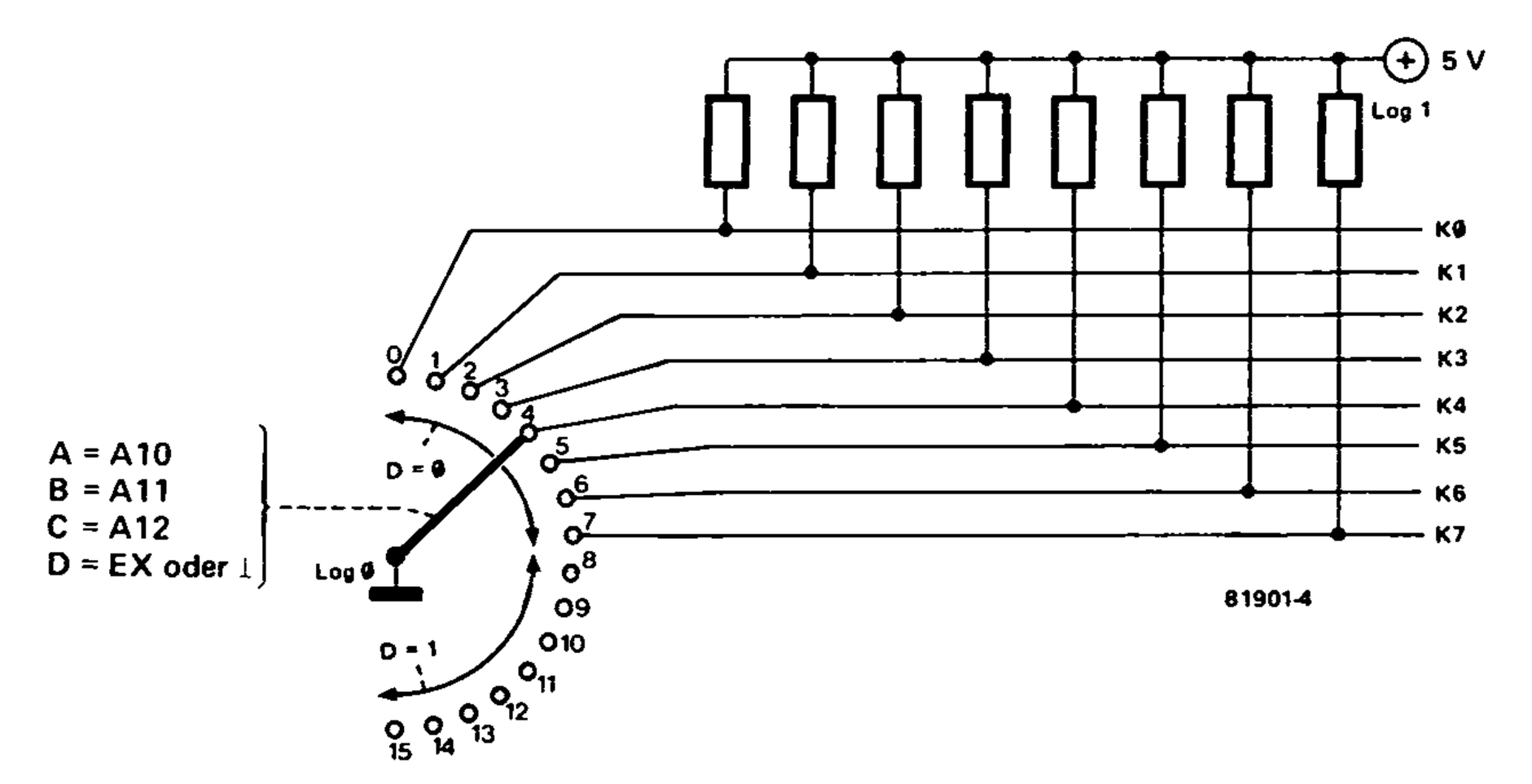


Bild 4. Eine vereinfachte Darstellung der Funktion von Adressendekoder IC6 auf der Basiskarte. Er gleicht einem Vielfachumschalter mit 16 Kontakten, von denen jedoch nur 8 Kontakte angeschlossen sind.

Tabelle 1. Wahrheitstabelle von Adressendekoder IC6 auf der Basiskarte.

D = ⊥ (φ) of D = EX = $\overline{8K0}$	C = A12	B = A11	A = A10	K7	K6	K5	K4	K3	K2	K1	K0	S = Standardkarte I = Interfacekarte B = Buskartenspeicher
0	0	0	0	1	1	1	1	1	1	1	0	S
0	0	0	1	1	1	1	1	1	1	0	1	I
0	0	1	0	1	1	1	1	1	0	1	1	I
0	0	1	1	1	1	1	1	0	1	1	1	I
0	1	0	0	1	1	1	0	1	1	1	1	I
0	1	0	1	1	1	0	1	1	1	1	1	I
0	1	1	0	1	0	1	1	1	1	1	1	S + I
0	1	1	1	0	1	1	1	1	1	1	1	S
1	0	0	0	1	1	1	1	1	1	1	1	B
1	0	0	1	1	1	1	1	1	1	1	1	B
1	0	1	0	1	1	1	1	1	1	1	1	B
1	0	1	1	1	1	1	1	1	1	1	1	B
1	1	0	0	1	1	1	1	1	1	1	1	B
1	1	0	1	1	1	1	1	1	1	1	1	B
1	1	1	0	1	1	1	1	1	1	1	1	B
1	1	1	1	1	1	1	1	1	1	1	1	B

Speichererweiterung

Die auf der Basisplatine vorhandenen Speicher-ICs werden zum einen durch weitere Speicher auf der Interfacekarte ergänzt; zum anderen können bei Bedarf noch zusätzliche Speicherkarten angeschlossen werden. Letztere werden am Schluß dieses Kapitels besprochen. Auf der Interfacekarte ist bereits so viel Platz für Speicher-ICs reserviert (maximal 5 K, selektiert durch K1 . . . K5), daß der Standard-Adressenbereich von 8 K ausgefüllt ist. Die RAMs IC2 und IC3 (2 x 2114) haben zusammen eine Kapazität von 1 K. Sie werden durch $\overline{CS}$ = K1 = 0 selektiert; die Adressen dieses 1-K-RAM lauten daher 0400 . . . 07FF.
Die hier gespeicherten Seiten 04 . . . 07 schließen unmittelbar an die Seiten 00 . . . 03 der Basisversion an; längere Anwenderprogramme können daher zusammenhängend gespeichert werden.
Für IC4 und IC5 lassen sich wahlweise 1-K-RAMs (8114), 1-K-EPROMs (2708) oder 2-K-EPROMs (2716) einsetzen. Diese Chips werden über die Gatter N42 . . . N44 von den Signalen K2 . . . K5 sowie vom Taktsignal Φ 2 aktiviert. Das Taktsignal Φ 2 sorgt für das richtige Timing beim Lesen und, sofern erforderlich, beim Schreiben in bzw. aus diesen Speicher-ICs.
Die Adressenbereiche von IC4 unf IC5 sind wie folgt festgelegt:
K2 = 0 Adressen 0800 . . . 0BFF 0,5 bzw. 1 x IC4
K3 = 0 Adressen 0C00 . . . 0FFF 0,5 bzw. 1 x IC4

K4 = Ø Adressen 1ØØØ . . . 13FF 0,5 bzw. 1 x IC5
K5 = Ø Adressen 14ØØ . . . 17FF 0,5 bzw. 1 x IC5
Setzt man für IC4 und IC5 1-K-RAMs oder 1-K-EPROMs ein, dann gibt es für beide ICs jeweils zwei identische Adressenbereiche. Bei den 2-K-EPROMs (2716) werden dagegen jeweils zwei K-Signale kombiniert: K2 & K3 für IC4 (Adressenbereich ØØØØ . . . ØFFF) und K4 & K5 für IC5 (Adressenbereich 1ØØØ . . . 17FF). Um die Signale K2 . . . K5 in Abhängigkeit von der gewählten IC-Bestückung an die Eingänge der Gatter N42 und N43 legen zu können, sind auf der Interfacekarte die Lötpunkte A . . . F angebracht. Dort müssen Drahtbrücken eingelötet werden, deren Lage in Tabelle 7 angegeben ist. Dies gilt auch für die Lötpunkte G . . . O und G' . . . O'; die Lage dieser Brücken, das sei hier schon vorweggenommen, hängt von den erforderlichen Speisespannungen sowie davon ab, ob zusätzlich die Adreßleitung AB1Ø oder die RAM-R/W-Leitung benötigt wird.

Vollständige oder unvollständige Adressendekodierung?

Die drei höchstwertigen Adreßleitungen A13 . . . A15 sind auf der Basisplatine nirgends angeschlossen, so daß ihre Signale bei der Adressierung keine Rolle spielen. Mit den übrigen 13 Leitungen des Adreßbus lassen sich 2^{13} Speicherplätze adressieren, was einer Kapazität von 8 K entspricht (1 K = 1024 Byte). Dies sind genau diejenigen 8 K, die über die Selektionssignale KØ . . . K7 des Adreßdekoders (siehe Bild 4) selektiert werden können. Es wurde bereits davon gesprochen, daß auf der Basiskarte und auf der Interfacekarte zusammen insgesamt 8 K untergebracht werden können. Solange keine weiteren Karten, die Busspeicherkarten, hinzugefügt werden, reichen daher die 13 bisher benutzten Adreßleitungen aus. Wird das System durch eine oder mehrere Busspeicherkarten erweitert, dann muß ein anderer Abschnitt des insgesamt 64 K umfassenden, vom 6502 adressierbaren Bereichs belegt werden. Dies bedeutet, daß auch die drei bisher unbenutzten Adreßleitungen A13 . . . A15 herangezogen werden müssen. Dazu ist folgendes notwendig:

1. Es muß sichergestellt sein, daß bei der Adressierung einer Busspeicherkarte nicht gleichzeitig ein anderer Baustein adressiert wird, der sich auf der Basis- oder auf der Interfacekarte befindet (siehe auch Tabelle 1). Das Lesen oder Schreiben von zwei Speicherplätzen zu gleicher Zeit muß auf jeden Fall verhindert werden, da dies zwangsläufig zur Verfälschung von Daten führt. Die Lösung dieses Problems ergibt sich unmittelbar aus Bild 4 und Tabelle 1: Eingang D von Adreßdekoder IC6 wird an EX gelegt (Drahtbrücke ändern); Signal EX muß immer dann logisch 1 sein, wenn eine Busspeicherkarte selektiert wird. Der Schalter in Bild 4 steht in diesem Fall in einer Position, bei der keine der Leitungen KØ . . . K7 an Masse liegt.
2. Auf den Busspeicherkarten müssen weitere Adressendekoder vorhanden sein, die die dort benötigten Selektionssignale erzeugen. Dies wird im Zusammenhang mit der RAM/EPROM-Karte am Schluß dieses Kapitels besprochen.

Woher nehmen wir nun ein geeignetes Signal EX, das den Eingang D von Adressendekoder IC6 steuert? Die Signale A13, A14 und A15 des Adreßbus werden einfach einem ODER-Gatter zugeführt. In Bild 1 (rechts

oben) ist dies das NOR-Gatter N33, zusammen mit dem als Inverter geschalteten Gatter N34. Sobald eine der drei Adreßleitungen logisch 1 wird, geht auch EX auf logisch 1. Oder anders ausgedrückt: Nur wenn die drei Adreßleitungen gleichzeitig Ø sind, ist EX = Ø. Damit ergeben sich folgende Adressenbereiche:
Alles, was adressierbar ist und sich auf der Basis- oder der Interfacekarte befindet (RAM, EPROM, PIA, VIA), ist erreichbar im Bereich
ØØØØ . . . 1FFF (Seiten ØØ . . . 1F).
Dies sind 32 Seiten oder 8 K.
Abhängig vom Umfang der zusätzlichen Speicherkapazität belegen die Busspeicherkarten teilweise oder vollständig den übrigen Bereich von 56 K. Dieser Bereich hat die Adressen
2ØØØ . . . FFFF (Seiten 2Ø . . . FF).
Solange die Signale A13, A14 und A15 des Adreßbus ohne Funktion sind (es sind keine Busspeicherkarten vorhanden), ist die Adressendekodierung unvollständig. Außerdem lassen sich dann alle Speicherplätze unter mehr als einer Adresse erreichen. Diese **Mehrfachadressierung** kommt so zustande:
Zu jeder Adresse gehören vier hexadezimale Ziffern (= Nibbles, halbe Bytes). Die ganz links stehende Ziffer, die gleichzeitig die höherwertige Ziffer der Seitenzahl ist, wird von den funktionslosen Signalen A13, A14 und A15 bestimmt. Da jedes dieser drei Adreßbussignale beliebig Ø und 1 werden kann, ohne daß sich dadurch irgendetwas ändert, ist jede Adresse mit sieben weiteren Adressen identisch. Daher wiederholt sich der Inhalt der Seiten ØX mit X = Ø . . . F auf den Seiten 2X, 4X, 6X, 8X, AX, CX und EX, während der Inhalt der Seiten 1X mit X = Ø . . . F auch auf den Seiten 3X, 5X, 7X, 9X, BX, DX und FX zu finden ist. Wenn dagegen das Signal EX an Eingang D von Adreßdekoder IC6 der Basisplatine liegt, ist damit eine notwendige Voraussetzung für die vollständige Adressendekodierung erfüllt; tatsächlich ist sie jedoch erst dann vollständig, wenn der bisher ungenutzte Bereich von 56 K mit Busspeicherkarten belegt wird. Unter den Adressen gibt es dann keine "Doppelgänger" mehr; jeder Speicherplatz hat jetzt notwendigerweise nur eine einzige Adresse.

Datenverkehrssteuerung

Das Signal EX gelangt unter dem Namen $\overline{8KØ}$ auch zu einem Adresseneingang von PROM IC17. In diesem PROM sind 32 Byte gespeichert; von jedem der aus 8 bit bestehenden Bytes sind hier jedoch nur die Bits Y1 (= $\overline{WRITE}$) und Y2 (= READ) von Bedeutung. Dies sind die Signale, die die Datenbuspuffer N17 . . . N32 steuern. Die Adressierung der 32 Byte erfolgt über die PROM-Eingänge A . . . E; die hier anliegenden Signale werden noch ausführlich besprochen.
Bild 5 erklärt die Funktion des PROM. Es arbeitet ähnlich wie ein Vielfachumschalter mit 32 Schalterstellungen. Die Position des Schaltarms bestimmt, welches der gespeicherten 32 Bitmuster zu den acht Datenausgängen durchgeschaltet wird. Die 32 Bitmuster (Byte) müssen vorher in das EPROM "eingebrannt" (programmiert) werden. Tabelle 2 gibt die Einbrennvorschrift in Form von Einsen und Nullen an. Bei diesem PROM erscheint übrigens das jeweils adressierte Byte an den Ausgängen, ohne daß

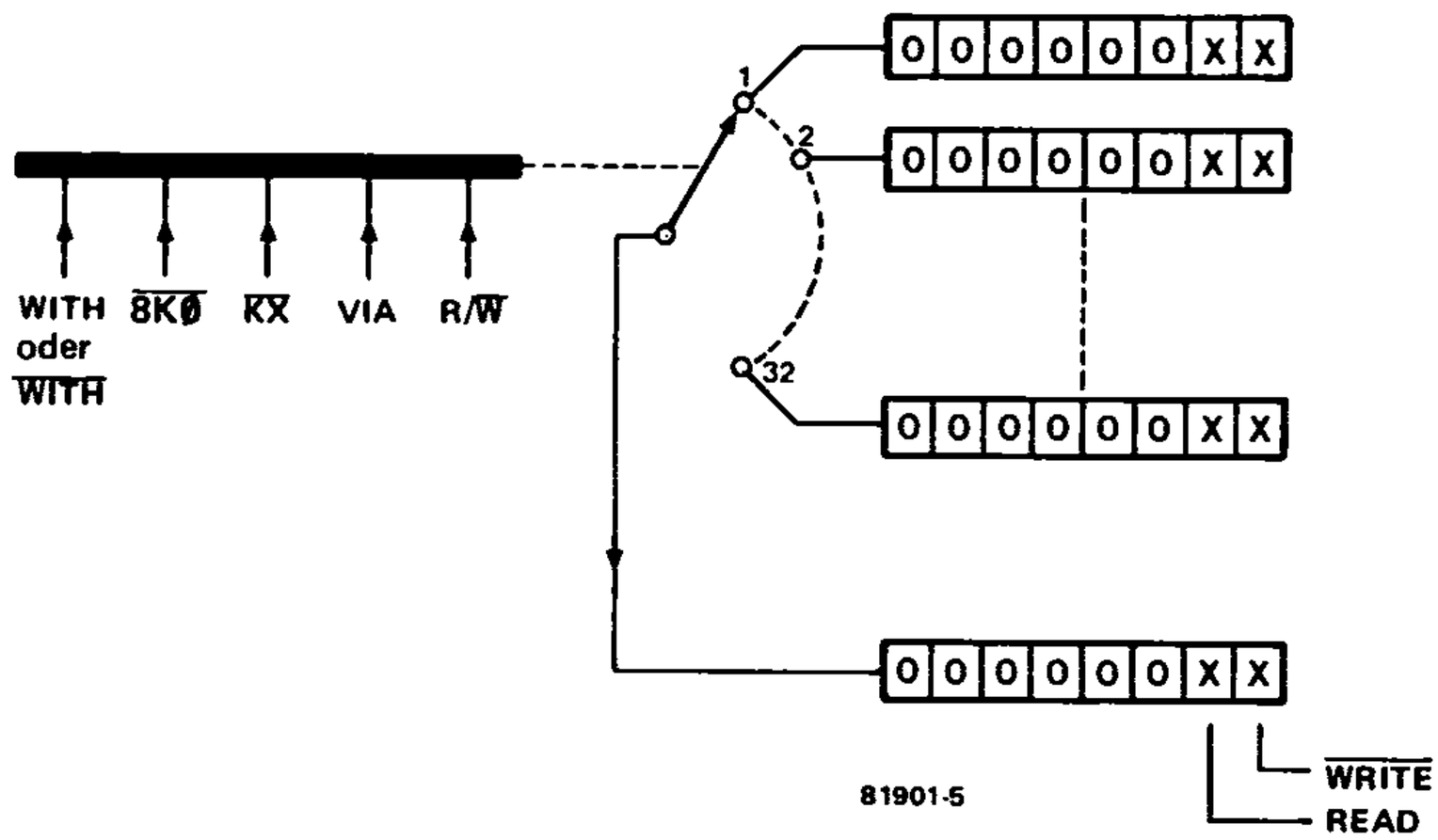


Bild 5. Das PROM IC17, dargestellt als Byte-Umschalter. Die Stellung des Schalters, die 32 Positionen einnehmen kann, wird von fünf verschiedenen Signalen gemeinsam bestimmt. Von den 8 bit, die abhängig von der Schalterstellung am Ausgang erscheinen, werden nur 2 bit weiterverwendet. Diese beiden Signale (WRITE und READ) steuern den Datenbuspuffer.

der Prozessor an diesem Lesevorgang direkt beteiligt ist.
Warum ist hier ein PROM notwendig? Schließlich muß dieses programmiert werden, und außerdem könnte man meinen, daß sich die Signale READ und WRITE für die Datenbuspuffer einfach aus dem Signal R/W des Prozessors ableiten lassen. Warum also umständlich, wenn es anscheinend auch einfach geht?
Aus zwei Gründen. Der **erste** ist folgender: Gesetzt den Fall, daß Daten aus einem der Bausteine gelesen werden sollen, die sich auf der Basiskarte des Junior-Computers befinden und deshalb am ungepufferten Datenbus liegen. Mit ihm sind auch die Ausgänge der Datenbuspuffer N18, N20 usw . . . N32 verbunden. Wenn das R/W-Signal des Prozessors die Puffer direkt steuern würde, wären diese Puffer jetzt aktiv. Der Datenbus würde von ihren Ausgangssignalen "beherrscht", so daß eine Verfälschung der zu lesenden Daten unvermeidlich wäre. Auf dem Datenbus würden nämlich infolge der niederohmigen Ausgänge der genannten Puffer nicht die gewünschten Daten, sondern irgendwelche anderen Signale erscheinen (siehe Bild 6). Um dies zu verhindern, müssen im beschriebenen Fall sämtliche 16 Datenbuspuffer blockiert werden, was einer Abtrennung aller (vom Prozessor aus gesehen) hinter den Puffern liegenden Bausteine gleichkommt. Beim **Schreiben** in Bausteine auf der Basiskarte sind die Dinge dagegen anders gelagert: Hierbei treten derartige Probleme nicht auf, da die nicht adressierten Bausteine die an ihren Dateneingängen ankommenden Signale ignorieren.
Noch ein zweiter Grund macht das PROM IC17 notwendig: Die Vektoren NMI, RES und IRQ (siehe Kapitel 3 in Buch 1) können bei vollständiger Adressendekodierung unter den echten Adressen FFFA . . . FFFF anstelle von 1FFA . . . 1FFF des Basisversion-EPROMs abgelegt werden. Möglich ist dies natürlich nur, wenn Seite FF tatsächlich zu einem (E)PROM

Tabelle 2. Inhalt von PROM IC17. Die Adressen und die zugehörigen Daten sind in hexadezimaler Form angegeben.

PROM-Adresse (hex)	E = WITH oder $\overline{WITH}$	D = $\overline{8K0}$	C = $\overline{KX}$	B = VIA	A = R/W	Y3 . . . . Y8	Y2 = READ	Y1 = $\overline{WRITE}$	PROM-Daten (hex)	
00	0	0	0	0	0	0	0	0	00	①
01	0	0	0	0	1	0	1	1	03	
02	0	0	0	1	0	0	0	0	00	
03	0	0	0	1	1	0	0	0	00	
04	0	0	1	0	0	0	0	1	01	②
05	0	0	1	0	1	0	0	1	01	
06	0	0	1	1	0	0	0	0	00	③
07	0	0	1	1	1	0	1	1	03	
08	0	1	0	0	0	0	0	0	00	
09	0	1	0	0	1	0	0	0	00	
0A	0	1	0	1	0	0	0	0	00	
0B	0	1	0	1	1	0	0	0	00	
0C	0	1	1	0	0	0	0	1	01	④
0D	0	1	1	0	1	0	0	1	01	
0E	0	1	1	1	0	0	0	0	00	
0F	0	1	1	1	1	0	0	0	00	
10	1	0	0	0	0	0	0	0	00	⑤
11	1	0	0	0	1	0	1	1	03	
12	1	0	0	1	0	0	0	0	00	⑥
13	1	0	0	1	1	0	0	0	00	
14	1	0	1	0	0	0	0	1	01	
15	1	0	1	0	1	0	0	1	01	
16	1	0	1	1	0	0	0	0	00	⑦
17	1	0	1	1	1	0	1	1	03	
18	1	1	0	0	0	0	0	0	00	
19	1	1	0	0	1	0	0	0	00	
1A	1	1	0	1	0	0	0	0	00	
1B	1	1	0	1	1	0	0	0	00	
1C	1	1	1	0	0	0	0	0	00	⑧
1D	1	1	1	0	1	0	1	1	03	
1E	1	1	1	1	0	0	0	0	00	
1F	1	1	1	1	1	0	0	0	00	

gehört. Ist Seite FF nicht mit einem (E)PROM belegt, so müssen die Vektoren NMI, RES und IRQ wie bisher aus dem EPROM der Basiskarte geholt werden. Das aber kann nur geschehen, wenn alle 16 Datenbuspuffer inaktiv sind, so daß die auf der Basiskarte nicht angeschlossenen Leitungen A13, A14 und A15 des Adreßbus keinen Einfluß haben. Auf der anderen Seite ist es zwingend notwendig, Seite FF mit einem diese Vektoren enthaltenden (E)PROM zu belegen, sobald eine oder mehrere Speicherkarten über die Buskarte auf die Interfacekarte gesteckt werden. Die Vektoren können dann nämlich nicht mehr ersatzweise aus Seite 1F

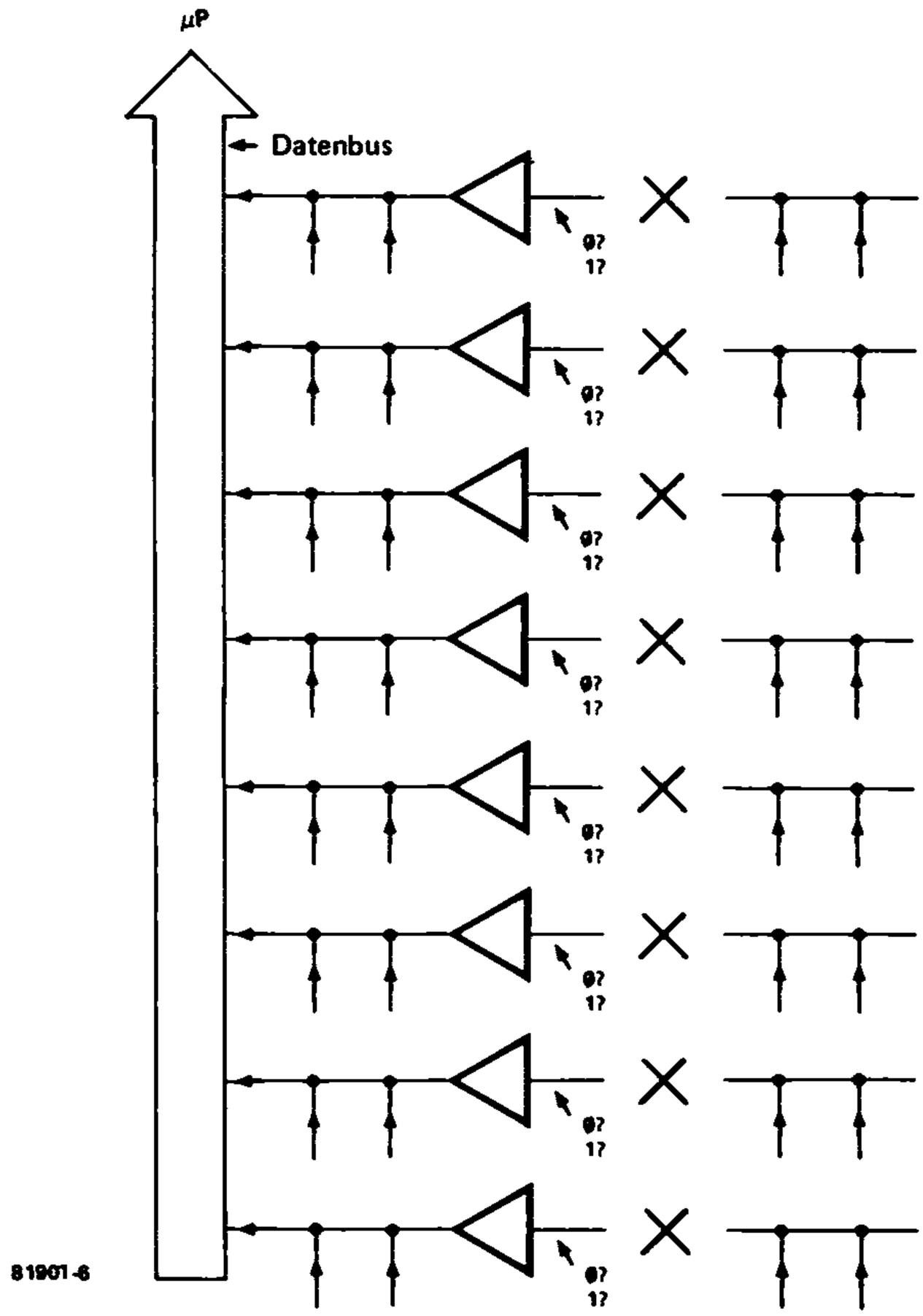


Bild 6. Wenn der Prozessor Daten aus einem Baustein auf der Basiskarte liest und der Datenbuspuffer nicht blockiert ist, können die zu lesenden Daten verfälscht werden. Im unblockierten Zustand sind nämlich die Ausgänge der Puffer niederohmig, so daß die von dort kommenden Signale den Datenbus "beherrschen".

gelesen werden, weil die Abtrennung der Buskarte durch die Puffer den ungehinderten Zugang zu den Speicherkarten verhindern würde.

Das PROM - Schaltstelle des Datenverkehrs

An den Adreßeingängen des PROM IC17 liegen folgende fünf Signale; sie nehmen Einfluß auf die Steuersignale $\overline{\text{WRITE}}$ und READ für die Datenbuspuffer:

A. Das Signal R/W oder genauer R/$\overline{\text{W}}$ (der Invertierungsstrich über dem W wurde bisher meistens weggelassen; er bedeutet: aktiver Zustand bei logisch Ø). Dieses Signal gibt an, ob eine Lese- oder eine Schreiboperation ausgeführt werden soll.

B. Das Signal VIA, das von N35 kommt und gleichzeitig als CS1-Signal für den VIA IC1 dient. Es ist logisch 1, wenn K6 und AB9 logisch Ø sind, also wenn der VIA aktiviert ist. Bei der Adressierung des 1-K-Bereichs, bei dem K6 = Ø ist, liegt der PIA vor und der VIA hinter den Datenbuspuffern.

C. Das Signal $\overline{KX}$. Dieses liefert das als Inverter arbeitende Gatter N36, das seinerseits von NAND-Gatter N40 gesteuert wird. $\overline{KX}$ ist logisch Ø, wenn eins der Signale K1 . . . K5 logisch Ø ist. Das aber ist bei der Adressierung eines Bausteins auf der Interfacekarte der Fall.

D. Das schon besprochene Signal $\overline{8KØ}$ = EX. Es ist logisch Ø, wenn der erste 8-K-Bereich (Speicher und I/O auf der Basiskarte und auf der Interfacekarte) adressiert wird, und logisch 1, wenn Speicherbausteine auf der Buskarte adressiert werden.

E. Eingang E von IC17 ist entweder mit der Speisespannung (Drahtbrücke R-S) oder mit Masse (Brücke R-T) verbunden. Dieser Eingang muß logisch Ø sein, wenn keine Buskartenspeicher vorhanden sind ($\overline{WITH}$ = WITHOUT = ohne); im anderen Fall liegt er an +5 V.

In Tabelle 2 ist der Inhalt von PROM IC17 angegeben. Um das PROM programmieren zu lassen, kann man den Elektor-Programmierservice in Anspruch nehmen. Näheres ist der jeweils aktuellen Ausgabe der Zeitschrift Elektor zu entnehmen. Betrachten wir Tabelle 2: Die Bits Y3 . . . Y8 sind ständig logisch Ø; sie spielen hier keine Rolle. Von den 32 möglichen Kombinationen der Eingangssignale A . . . E werden nur acht benutzt, da die Datenbuspuffer in Abhängigkeit von acht unterschiedlichen Situationen gesteuert werden müssen:

1. Schreiben oder Lesen in bzw. aus Bausteinen auf der Interfacekarte (IC2 . . . IC5). Es sind die acht Datenbuspuffer aktiviert, in deren Richtung gelesen bzw. geschrieben werden soll (PROM-Adressen ØØ und Ø1).
2. Lesen aus dem EPROM und Lesen oder Schreiben aus bzw. in das RAM oder den PIA auf der Basiskarte. Sämtliche Datenbuspuffer sind blokkiert (PROM-Adressen Ø4 und Ø5).
3. Schreiben oder Lesen des VIA. Er befindet sich auf der Interfacekarte, so daß die acht Datenbuspuffer aktiviert werden müssen, die in die Schreib- bzw. in die Leserichtung zeigen (PROM-Adressen Ø6 und Ø7).
4. Schreiben oder Lesen von Adressen, die bei vollständiger Adressendekodierung auf den Busspeicherkarten liegen würden; Busspeicherkarten sind jedoch (wie bei 1. bis 3.) nicht vorhanden. Eingang E von IC17 wurde wie beschrieben an logisch Ø gelegt. Die in Leserichtung zeigenden Datenpuffer werden blockiert; die Vektoren NMI, RES und IRQ werden automatisch aus dem EPROM (2708) auf der Basisplatine geholt (PROM-Adressen ØC und ØD).
5. Wie 1., jedoch Busspeicherkarten vorhanden (PROM-Adressen 1Ø und 11).
6. Wie 2., jedoch Busspeicherkarten vorhanden (PROM-Adressen 14 und 15).
7. Wie 3., jedoch Busspeicherkarten vorhanden (PROM-Adressen 16 und 17).
8. Wie 4., jedoch Busspeicherkarten vorhanden. In diesem Fall (Eingang E von PROM IC17 liegt an logisch 1) müssen die betreffenden Datenbus-

puffer beim Lesen aktiviert werden. Die Vektoren NMI, RES und IRQ werden wie beschrieben aus Seite FF geholt; diese Seite muß deshalb mit einem (E)PROM belegt sein, das unter den Adressen FFFA . . . FFFF diese drei Vektoren enthält:

FFFA = $2F
FFFB = $1F
FFFC = $1D
FFFD = $1C
FFFE = $32
FFFF = $1F

Von den 32 in PROM IC17 gespeicherten Bytes wird nur die Hälfte verwertet, da die zur anderen Hälfte gehörenden Kombinationen von D = $\overline{8K\emptyset}$, C = $\overline{KX}$ und B = VIA normalerweise nicht auftreten. Doch auch wenn das geschehen würde, bliebe es unschädlich: Da hier ausschließlich Nullen im PROM stehen, zeigen in einem solchen Fall die aktiven Datenbuspuffer in Schreibrichtung; eine Verfälschung der Datenbussignale ist somit ausgeschlossen.

Zusammenfassend läßt sich feststellen, daß die Datenbuspuffer in den folgenden beiden Fällen blockiert sind:

a. Es wird ein Baustein auf der Basiskarte adressiert.
b. Es werden Vektoren gelesen, die eigentlich auf Seite FF stehen müßten. Da diese Seite nicht belegt ist, muß ersatzweise auf Seite 1F (EPROM auf der Basiskarte) gelesen werden.

In drei Fällen ist dagegen ein Datenaustausch über die Puffer möglich:

c. Es wird ein Speicherbaustein auf der Interfacekarte adressiert.
d. Der VIA auf der Interfacekarte wird adressiert.
e. Es wird ein Baustein auf den Busspeicherkarten adressiert, beispielsweise zum Lesen der Vektoren auf Seite FF.

Damit ist die Besprechung der Schaltung in Bild 1 abgeschlossen. Wenden wir uns nun Bild 2 zu.

Kassetten-Hardware

Der größte Teil der Schaltung in Bild 2 gehört zum Kassetteninterface, das zusammen mit der zugehörigen Software "TAPE MANAGEMENT" (siehe Kapitel 11) das Speichern von Daten auf normalen Tonbandkassetten ermöglicht. Dazu müssen die zu speichernden Daten vorher in Töne umgesetzt werden; umgekehrt ist beim Lesen der Daten die Rückwandlung der aufgezeichneten Tonsignale in die vom Computer lesbare Datenform notwendig. Wie dies im einzelnen geschieht, soll nun betrachtet werden.

Als Datenleitung zwischen Prozessor und Kassetteninterface wird die Leitung PB7 des Portkonnektors benutzt. Während der Aufzeichnung von Daten auf Band ist PB7 ebenso wie PB5 als Ausgang programmiert. Bei dieser Operation liegt PB5 ständig auf logisch Ø und PB6 ständig auf logisch 1. Die Ø am linken Eingang (Pin 8) von Gatter N38 hat eine 1 an seinem Ausgang zur Folge; dieser Ausgang mit offenem Kollektor ist dann hochohmig. Infolge der Ø an Leitung PB5 leitet der PNP-Darlingtontransistor T3, so daß der Kontakt von Relais Re2 geschlossen ist und gleichzeitig die rote LED (OUTPUT ON) aufleuchtet. Der Relaiskontakt verbindet die beiden Anschlüsse von Buchse J4 und schaltet über den Fern-

bedienungsanschluß den Motor des mit OUTPUT bezeichneten Kassettenrekorders ein. Vorausgesetzt, daß bei diesem Rekorder die Aufnahmetaste gedrückt ist, läßt sich so der Aufnahmevorgang durch Software starten und stoppen.
Die während des Schreibens von Daten auf Band an PB6 liegende logisch 1 sorgt dafür, daß T2 sperrt und dadurch Relais Re1 sowie die grüne LED stromlos sind. Der Motor des Kassettenrekorders INPUT ist daher ausgeschaltet.
Über Leitung PB7, Inverter N39 und Buchse J2 gelangen die bereits mit Hilfe von Software in Tonsignale umgewandelten Datenbits zum Eingang des Kassettenrekorders OUTPUT. Sie passieren dabei das Trimmpoti P2, mit dem sich der Pegel der tonfrequenten Spannung einstellen läßt.
Sollen Daten von einer Kassette gelesen werden, dann wird PB7 als Eingang programmiert, PB5 und PB6 bleiben dagegen als Ausgang geschaltet. Jetzt ist jedoch PB5 logisch 1 und PB6 logisch Ø, so daß das Ausgangssignal von Komparator IC7 über Gatter N38 auf die Leitung PB7 gelangen kann. Dieses Signal wird auch von Gatter N39 weitergeleitet, was jedoch bedeutungslos ist.
Da PB5 logisch 1 ist, bleibt der an Buchse J4 liegende Relaiskontakt offen; die rote LED leuchtet nicht auf. PB6 ist dagegen Ø: nun leitet T2, so daß Relais Re1 anzieht und die grüne LED (INPUT ON) aufleuchtet. Über Buchse J3 kann der Strom des Antriebsmotors von Kassettenrekorder INPUT fließen.
Es ist nicht notwendig, zwei getrennte Kassettenrekorder für das Lesen und Schreiben von Daten zu verwenden. Steht nur ein Kassettenrekorder zur Verfügung, dann werden die Kontakte der beiden Relais einfach parallelgeschaltet. Man kann aber auch ohne Relais, mit "Handbetrieb" arbeiten, so daß beide Relais entfallen. Die visuelle Information über die Betriebsart, die die beiden LEDs liefern, hat sich übrigens in der Praxis als recht nützlich erwiesen. Auch hier steht die Farbe Rot für "Achtung, Aufnahme!", während die grüne LED "Freie Fahrt" für die Daten vom Band zum Computer signalisiert.

Die PLL: ein Frequenzfolger als Bitmacher

Der Teil der Schaltung in Bild 2, der zwischen Buchse J1 und Gatter N38 liegt, wandelt die vom Kassettenrekorder INPUT kommenden Tonsignale in computergerechte Signale zurück. Kennern des schon vor Jahren erschienenen Einplatinencomputers KIM wird die Schaltung mit IC6 und IC7 sicherlich nicht unbekannt vorkommen. Dieser Eindruck ist richtig! Hier wurde auf Altbewährtes zurückgegriffen; lediglich einige kleinere Verbesserungen ergänzen die KIM-Kassetteninterface-Hardware.
Die Daten sind in der Weise auf dem Band gespeichert, daß Töne mit zwei unterschiedlichen Frequenzen einander abwechseln: Auf einen Ton mit einer Frequenz von ca. 3600 Hz folgt ein Ton von ca. 2400 Hz, danach wieder ein 3600-Hz-Ton gefolgt von einem 2400-Hz-Ton usw. Jedes aus einem 3600-Hz-Ton und einem 2400-Hz-Ton bestehende Tonpaar repräsentiert ein Bit. An der Zeitdauer des 3600-Hz-Tons in Relation zur Länge des 2400-Hz-Tons erkennt der Prozessor (mit Unterstützung der Interface-Software; siehe Kapitel 11), ob dieses Bit eine logische 1 oder eine Ø ist.

In Bild 7 sind sowohl die beiden Tonsignale für logisch Ø und 1 (Signale ① und ②) als auch die Signale skizziert, die daraus mit Hilfe von IC6 und IC7 entstehen (Signale ③ und ④). Die Schaltung mit IC6 und IC7 setzt lediglich ein an Eingang J1 liegendes Tonsignal mit einer Frequenz von 3600 Hz in logisch 1 und ein 2400-Hz-Tonsignal in logisch Ø um. IC6 (ein 565) ist ein sogenanntes PLL-IC (PLL = Phase Locked Loop = phasenverriegelte Schleife). Über die Arbeitsweise derartiger Systeme ließe sich ein weiterer Band der Junior-Computer-Buchreihe füllen; die höhere Mathematik würde darin eine wichtige Rolle spielen. Im Zusammenhang mit dem Kassetteninterface des Junior-Computers sollen jedoch einige beschreibende Sätze genügen:
Das PLL-IC kann hier von außen als *Frequenzfolger* betrachtet werden. Ebenso wie bei einem Emitterfolger die Ausgangsspannung der Eingangsspannung folgt, paßt sich bei dem PLL ein interner Oszillator mit seiner Frequenz (innerhalb bestimmter Grenzen) an die Frequenz des Eingangssignals an. Voraussetzung ist natürlich, daß das PLL-Eingangssignal weder zu schwach noch zu stark ist. Die Frequenz des PLL-internen Oszillators wird von einer Gleichspannung gesteuert; der Oszillator ist daher ein VCO (Voltage Controlled Oscillator). Hier beträgt die Oszillatorfrequenz bei fehlendem Eingangssignal ca. 3000 Hz, so daß sie etwa in der Mitte zwischen 2400 Hz und 3600 Hz liegt. Hat das Eingangssignal eine Frequenz von 3600 Hz, dann steigt die VCO-Frequenz um 600 Hz an, bei 2400 Hz am Eingang sinkt sie um 600 Hz ab. Um die VCO-Frequenz herabzusetzen, ist eine höhere Steuerspannung notwendig, während die VCO-Frequenz bei niedrigerer Steuerspannung ansteigt. Die Höhe der im PLL-IC selbsttätig erzeugten VCO-Steuerspannung ist daher ein Unterscheidungsmerkmal für die Frequenz des Eingangssignals. Durch Vergleich mit einer Referenzspannung läßt sich eine logische Aussage darüber treffen, ob die Eingangsfrequenz über oder unter 3000 Hz liegt. Man nennt dieses Verfahren der Darstellung von digitalen Signalen durch Tonsignale auch Frequency Shift Keying, oder abgekürzt FSK.

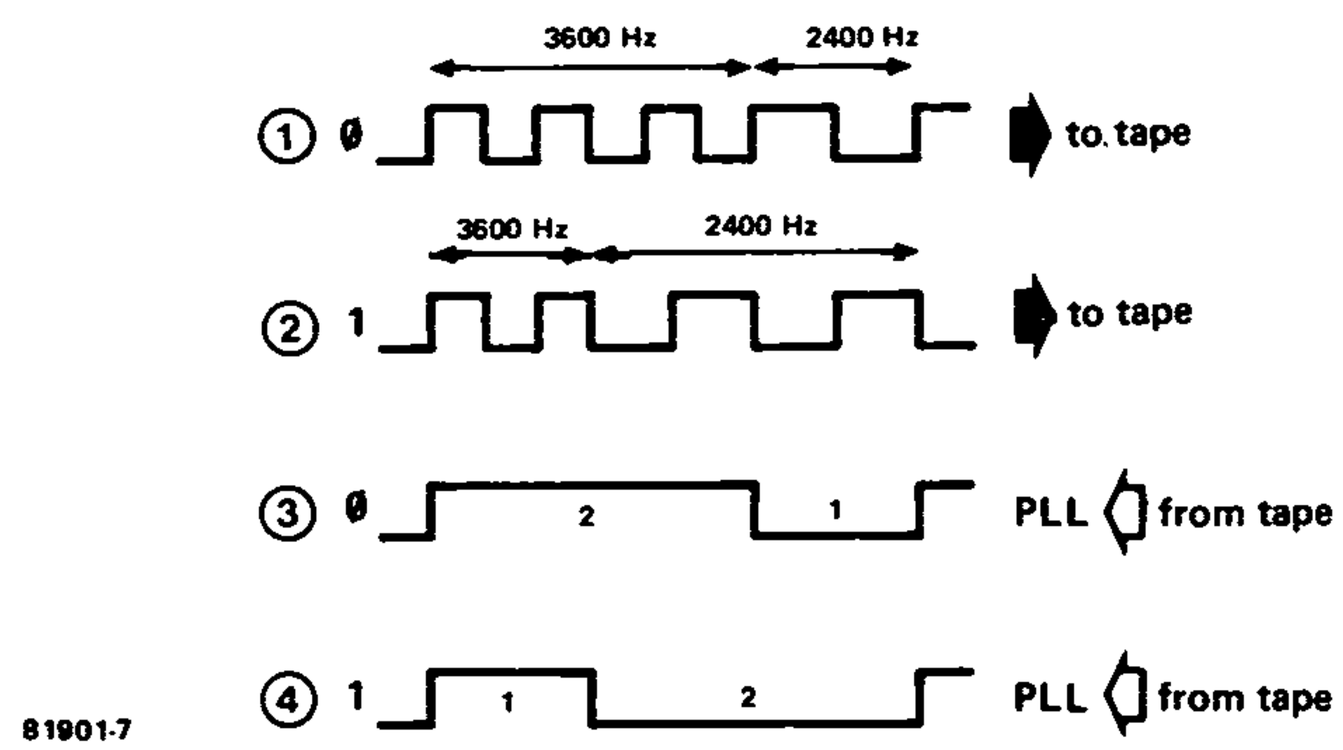


Bild 7. Die logischen Signale Ø und 1 werden vom Kassetteninterface in tonfrequente Impulsreihen "übersetzt", bevor sie zum Rekorder gelangen (① und ②). Beim Lesen der Daten vom Band erzeugt eine PLL-Schaltung aus den nun nicht mehr ganz rechteckförmigen Tonsignalen die Signale ③ und ④, die vom Computer mit Hilfe des Systemprogramms TM weiterverarbeitet werden.

Der 565 (IC6) wird mit Speisespannung über die beiden Dioden D2 und D3 versorgt, denen Kondensator C8 parallelgeschaltet ist. Dieses Siebglied unterdrückt wirkungsvoll Störimpulse auf der Speisespannung, die den Betrieb des PLL beeinträchtigen können. Die geglättete Speisespannung an Pin 10 von IC6 wird auch zur Gleichspannungseinstellung der Eingänge (Pin 2 und 3) herangezogen. Der KIM benutzt hierzu eine ungesiebte 5-V-Speisespannung, obwohl auch dort eine geglättete Spannung von 12 V vorhanden ist! Ferner wird hier im Gegensatz zum KIM das Eingangssignal nicht um den Faktor 10 abgeschwächt, bevor es den IC-Eingang (Pin 2) erreicht. Die genannten, nicht gerade aufwendigen, aber trotzdem sehr wirkungsvollen Verbesserungen führen zu einer hohen Störsicherheit gegen "Dropouts" auf dem Band und Verschmutzung oder falscher Einstellung des Tonkopfs.
Von den beiden Eingängen des 565 (Pin 2 und 3) wird nur einer vom Tonsignal gesteuert. Die beiden Eingänge gehören zu einem internen Differenzverstärker; hier liegt also asymmetrische Steuerung vor. Koppelkondensator C1 hat eine wesentlich niedrigere Kapazität als der entsprechende Kondensator des KIM, so daß hier Störspannungen mit niedrigen Frequenzen (z.B. Netzbrummen) besser unterdrückt werden. Widerstand R37 ist beim Anschluß von Buchse J1 an den Lautsprecher- oder Kopfhörerausgang eines Kassettenrekorders als Last bei abgeschaltetem Lautsprecher notwendig.
Die VCO-Frequenz ohne Eingangssignal wird von C3, R25 und P1 bestimmt. Trimmpoti P1 ist sorgfältig auf die Mittenfrequenz von 3000 Hz einzustellen; hiervon hängt die Zuverlässigkeit der Datenübertragung maßgeblich ab. Später werden noch mehrere Methoden angegeben, die sicher zum Ziel führen.
Pin 7 von IC6 ist der Ausgang des PLL. An diesen Pin ist die Steuerspannung des internen VCO herausgeführt. Kondensator C9, der zwischen dem Ausgang und +12 V liegt, bildet zusammen mit einem IC-internen Widerstand von 3k6 das sogenannte Schleifenfilter. Von der Dimensionierung des Schleifenfilters hängt das dynamische Verhalten des PLL bei Frequenzänderungen des Eingangssignals ab. Das Ausgangssignal von IC6 gelangt über den Tiefpaß R26 . . . R28/C5 . . . C7 zum invertierenden Eingang des Komparators IC7. Sein nichtinvertierender Eingang liegt über R29 an einer ebenfalls von IC6 gelieferten festen Gleichspannung. Die Dimensionierung des Tiefpasses steht in Zusammenhang mit der Folgegeschwindigkeit der 3600- und 2400-Hz-Töne, die ihrerseits von der Aufzeichnungsgeschwindigkeit der Daten auf die Kassette abhängt. Diese Geschwindigkeit, auch Baudrate genannt, beträgt beim Junior-Computer 800 Baud; auf diesen Wert ist sowohl die Kassettenhardware als auch die Kassettensoftware abgestimmt.
Wie schon beschrieben hat eine hohe Eingangsfrequenz (3600 Hz) eine niedrige Gleichspannung am Ausgang des PLL zur Folge, bei niedriger Eingangsfrequenz (2400 Hz) ist die PLL-Ausgangsspannung dagegen hoch. Da am nichtinvertierenden Eingang von Komparator IC7 die mittlere Spannung liegt und sein Ausgang über R31 mit +5 V verbunden ist, erscheint an letzterem ein TTL-gerechtes logisches Signal: Der Ausgang von IC7 ist logisch 1 bei 3600 Hz an J1 und logisch Ø bei 2400 Hz an dieser Buchse. Das Signal wird von Gatter N38 invertiert und anschließend auf

Leitung PB7 gesetzt.
Noch eine letzte Bemerkung zur Kassetteninterface-Hardware: Der PLL kann nicht unendlich schnell auf Frequenzänderungen des Eingangssignal reagieren. Die Folge ist, daß der Ausgang von Komparator IC7 beim Wechsel von 3600 Hz nach 2400 Hz oder umgekehrt nicht einfach von einem logischen Zustand zum anderen springt. Es pendelt kurzzeitig zwischen beiden hin und her, bevor der richtige Zustand stabil bleibt. Dieser sogenannte PLL-Jitter, der weitgehende Ähnlichkeit mit dem Prellen eines mechanischen Kontakts hat, wird von der Kassetteninterface-Software (siehe Kapitel 11 sowie Buch 4) unterdrückt.

RS-232-Peripherie

ein Tor zur Außenwelt

Der Teil der Schaltung in Bild 2, der noch nicht besprochen wurde, ist zwar weniger umfangreich, ihm fällt aber trotzdem eine wichtige Rolle zu. T1, N37 und IC8 sorgen zusammen mit einigen passiven Bauelementen dafür, daß Daten mit peripheren Einheiten, zum Beispiel mit dem Elekterminal (Datensichtgerät mit alfanumerischer Eingabetastatur), ausgetauscht werden können. Port PA7 ist der Eingang für ankommende Daten, während über PBØ Daten nach außen gesendet werden.
Transistor T1 arbeitet als Empfänger für die ankommenden Daten. Anzumerken ist dazu nur, daß er das Eingangssignal invertiert. Gesendet werden Daten von IC8: Da Pin 6 von Gatter N37 über R8 an +5 V liegt (Verbindung P-Q sei offen), invertiert N37 das Signal von PBØ. Der Ausgang von N37 steuert den nichtinvertierenden Eingang von Komparator IC8, der dieses Signal mit der am Verbindungspunkt von R11 und R12 liegenden Spannung vergleicht. Die Spannung am Komparatorausgang beträgt +12 V, wenn der Ausgang von N37 logisch 1 ist; bei einer logischen Ø am Gatterausgang beträgt sie −12 V. Abgesehen von der Verschiebung der Spannungen für logisch Ø und 1 findet auch zwischen PBØ und dem Ausgang von IC8 eine Invertierung des Signals statt.
Der Eingang des Datenempfängers und der Ausgang des Datensenders sind mit einem genormten 25-poligen D-Konnektor verbunden; er wird auch RS-232-Konnektor genannt. Die Norm RS-232 legt die Verfahrensweise für den Austausch von Daten fest: Die Daten werden seriell (Bit für Bit) übertragen, wobei jedem Byte ein Startbit vorangeht. Die Spannung, die logisch Ø repräsentiert, muß zwischen +5 V und +15 V liegen; für logisch 1 ist ein Spannungsbereich von −5 V bis −15 V vorgeschrieben (Version RS-232-C). Beim Junior-Computer betragen diese Spannungen ca. +12 V und −12 V. Eine logische 1 wird somit durch eine niedrige Spannung und eine Ø durch eine hohe Spannung dargestellt. Diese negative Logik bedingt, daß das ankommende und das abgehende Signal im Computer invertiert werden müssen, sofern man nicht mit "gespiegelten" Daten arbeiten will.
Am D-Konnektor, dessen Stiftbelegung von der Norm RS-232 ebenfalls festgelegt ist, sind zwischen bestimmten Stiften Brücken angebracht. Welche Kontakte miteinander verbunden werden müssen, hängt von der Art des angeschlossenen Geräts ab. Wie schon erwähnt, kommt hier vor allem das Elekterminal in Frage, das ursprünglich für das Elektor-SC/MP-

Mikrocomputersystem entwickelt wurde. Dieses Datensichtgerät ist auch für den Junior-Computer hervorragend geeignet.

Damit ist die Besprechung der Hardware auf der Interfacekarte (Bild 1 und 2) abgeschlossen. Bevor der erweiterte Junior-Computer in Betrieb genommen werden kann, müssen wir uns noch einmal der Basiskarte und der Speisung zuwenden. Einerseits muß nämlich die Speisung dem erhöhten Leistungsbedarf angepaßt werden, andererseits sind auch auf der Basiskarte einige kleinere Änderungen notwendig: hier oder da ein anderer Widerstandswert, eine umgelegte Drahtbrücke und eventuell ein Stückchen zusätzliche Elektronik im Briefmarkenformat.

Basiskarte und Speisung angepaßt

Wie bei einer Operation am menschlichen Körper Blut fließt, so fließen bei Änderungen an der Basis- und der Speiseplatine mehr oder weniger viele Lötzinntropfen. Zwar hätten wir von vornherein alles so gestalten können, daß bei Erweiterungen keine Änderungen oder Ergänzungen an bereits Bestehendem notwendig sind. Doch hier verhält es sich ähnlich wie beim Bau eines Hauses: Plant man den Ausbau des Dachgeschosses erst für einen späteren, noch unbestimmten Zeitpunkt, so wird man nicht sofort Löcher im Dach für die Mansarden aussparen lassen. Alle nachfolgend beschriebenen Modifikationen sind nämlich nur dann notwendig, wenn der Junior-Computer vom Ein- zum Mehrplatinencomputer wachsen soll.
Auf der **Basisplatine** werden drei Änderungen vorgenommen:

1. **Anpassung einiger Widerstände.** Während die Daten- und die Adreßleitungen auf der Interfacekarte gepuffert werden, ist dies bei den Leitungen des Steuerbus nicht der Fall. Eine Art Pufferung ist auch nur bei der R/W-Leitung notwendig: Der Ausgang von Gatter N6 auf der Basisplatine, der das Signal RAM-R/W liefert, erhält durch Herabsetzen des Werts von Pull-Up-Widerstand R5 auf 470 Ω die gewünschte höhere Leistungsfähigkeit. Die Schaltzeiten beim Wechsel der logischen Signale werden dadurch verkürzt. In gleicher Weise förderlich ist auch das Herabsetzen der Werte von R14, R15 und R16 (Leitungen KØ, K6 und K7) auf 470 Ω.

2. **Verlegen der Drahtbrücke** auf der Basisplatine von Masse nach EX. Wie bereits ausführlich beschrieben, ist diese Änderung nur beim Anschluß von Busspeicherkarten erforderlich. Wenn außer der Interfacekarte keine weiteren Karten hinzukommen, muß Eingang D von IC6 wie bisher mit Masse verbunden bleiben!

3. **Ergänzen der Schaltung um Gatter N5,** da das Ausführen von Programmen in Einzelschritten (Step-by-Step) beim erweiterten Junior-Computer in mehr als einem Fall blockiert werden muß. Bild 8a zeigt die bisherige und Bild 8b die neue Schaltung. In Bild 9 sind die beteiligten Signale in Form von Impulsdiagrammen dargestellt.
Wie aus der Instruktionsübersicht am Schluß von Buch 1 bekannt ist, nimmt die Ausführung jeder Instruktion ein bestimmtes Vielfaches einer

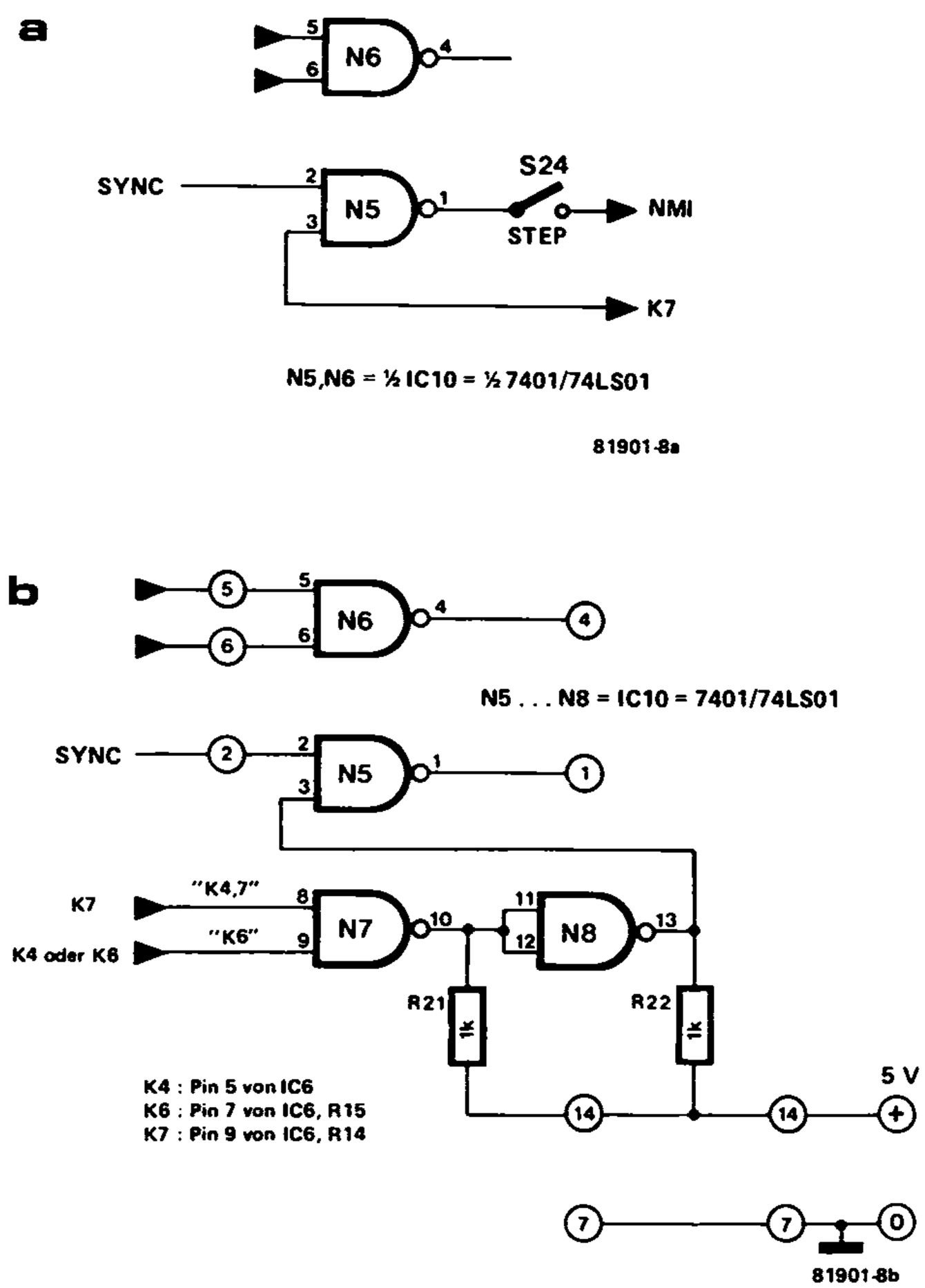


Bild 8. Ursprüngliche (a) und neue Schaltung (b) von IC10 auf der Basiskarte. Die Gatter sorgen dafür, daß der eingeschaltete Step-by-Step-Modus, das Abarbeiten eines Programms in langsamen Einzelschritten, in bestimmten Fällen automatisch aufgehoben wird. Notwendig ist dies beim Monitor und beim Systemprogramm PM (siehe Anhang 2).

Taktperiode an Zeit in Anspruch. In der letzten Ausführungsphase der vorangegangenen Instruktion wird der Opcode der nächsten Instruktion aus dem Speicher geholt. In dieser Phase ist das Signal R/W (② in Bild 9) logisch 1, da der Opcode gelesen werden muß. Gleichzeitig wird auch ein SYNC-Impuls (③ in Bild 9) erzeugt. Bei der bisherigen Schaltung (Bild 8a) hat der SYNC-Impuls einen NMI-Impuls am Ausgang von Gatter N5 zur Folge, es sei denn, daß Signal K7 logisch Ø ist. Im Step-by-Step-Modus entsteht daher nur dann ein NMI, solange nicht das EPROM auf der Basisplatine adressiert ist. Nach Auftreten eines NMI wird nur noch die Instruktion ausgeführt, deren Opcode bereits gelesen wurde.

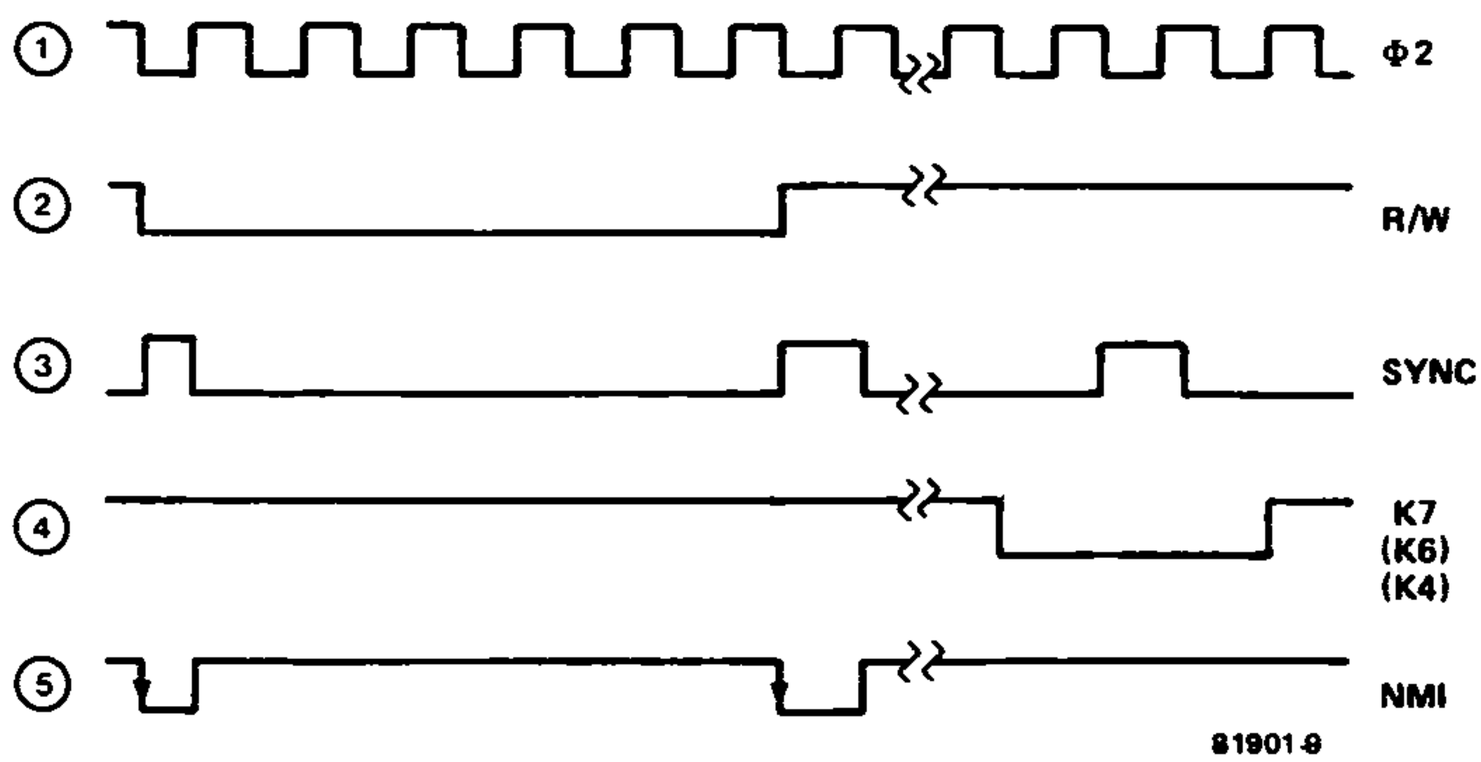


Bild 9. Während der Ausführungsphase eines Befehls, in der der Opcode aus dem Speicher gelesen wird, erzeugt der Prozessor einen SYNC-Impuls. Dieser kann in einigen Fällen einen Interrupt (NMI) auslösen. Der gerade laufende Befehl wird jedoch noch vollständig ausgeführt.

Unter der Voraussetzung, daß der NMI im Step-by-Step-Modus auf den Monitor (Startadresse 1CØØ) gerichtet ist, wird die Programmausführung in Einzelschritten bei jedem Eintritt in den Monitor aufgehoben. Im Monitor ist der Step-by-Step-Modus nämlich nicht erlaubt, da verschiedene Routinen (Displaysteuerung, Keyboard-Abfrage) ununterbrochen durchlaufen werden müssen.
Auch das Programm PRINTER MONITOR aus Kapitel 12 macht eine Aufhebung des Step-by-Step-Modus notwendig. Das gleiche gilt für solche Programme, die von Hand in das PIA-RAM eingegeben wurden und in Kombination mit dem Monitor (siehe Anhang 2) oder mit PM laufen. Das Programm PM wird von Signal K4 (in diesem Fall identisch mit K5) selektiert, während das Signal K6 das PIA-RAM anspricht. Mit der Schaltungserweiterung in Bild 8b wird erreicht, daß in den genannten Fällen kein NMI ausgelöst werden kann.
Bis auf zwei Widerstände ist die Hardware in Bild 8b auf der Basisplatine bereits vorhanden: nur zwei der vier Gatter von IC10 wurden bisher benutzt. IC10 wird aus seiner Fassung genommen; an seine Stelle tritt die Miniplatine in Bild 11, auf der IC10 seinen neuen Platz findet.
An dieser Stelle sei noch einmal nachdrücklich betont: **Die Platine in Bild 11 und die Widerstände R21 und R22 sind nur dann erforderlich, wenn mit dem PRINTER MONITOR ein Programm im Step-by-Step-Modus durchlaufen wird. Auch um dezimale Rechenoperationen schrittweise ausführen zu können (siehe Anhang 2), ist diese Hardware-Erweiterung notwendig.**

Höherer Strombedarf

Durch die Erweiterung des Junior-Computers mit der Interfacekarte und einer oder mehreren Busspeicherkarten wächst auch der Strombedarf. Insbesondere die RAMs sind relativ gefräßige Verbraucher. Daneben wird für das RS-232-Interface (siehe Bild 2) zusätzlich die Spannung −12 V

benötigt. Bild 10 zeigt die Schaltung der angepaßten Speisung. Hier ist im Vergleich zur bisherigen Ausführung einiges beim alten geblieben, einiges wurde geändert und einiges ist neu hinzugekommen. Die Speisung liefert in ihrer erweiterten Version folgende Spannungen und Ströme:

+5 V, maximal 4 A
−5 V, maximal 400 mA
+12 V, maximal 400 mA
−12 V, maximal 400 mA

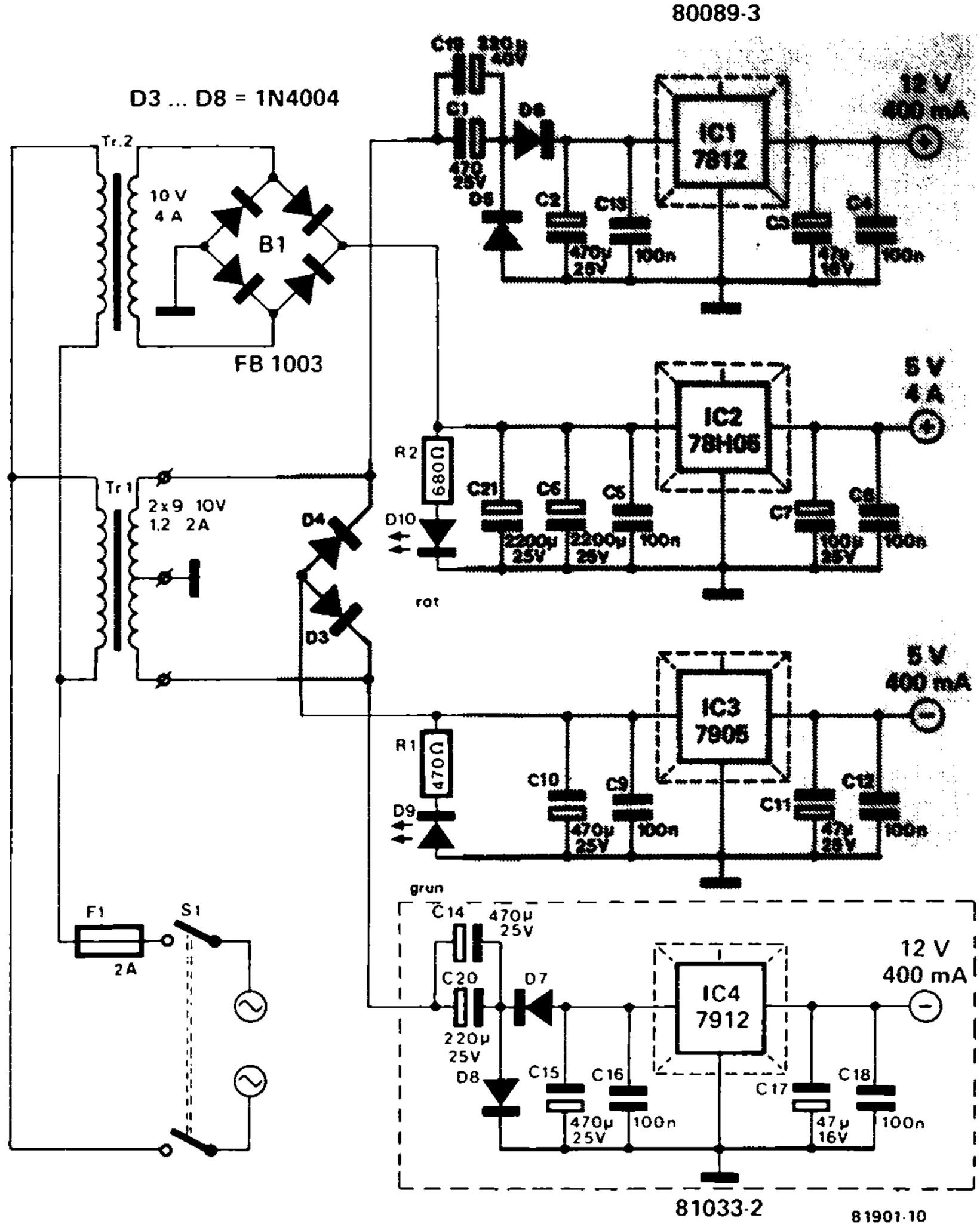


Bild 10. Schaltung des verstärkten Netzteils für den erweiterten Junior-Computer. Der grau unterlegte Schaltungsteil befindet sich auf der bereits vorhandenen Platine, der gestrichelt umrandete Teil (−12 V-Speisung) wird auf der Platine in Bild 13 untergebracht.

Dies reicht mit Sicherheit für den Betrieb der Basiskarte, der Interfacekarte und fünf voll bestückten Busspeicherkarten aus. Die externen Geräte des Junior-Computer-Systems, wie zum Beispiel das Elekterminal, sind normalerweise mit einer eigenen Speisung ausgestattet. Das Elekterminal kann jedoch bei Bedarf ebenfalls von der erweiterten Speisung versorgt werden (siehe Kapitel 12).

So viel graue Theorie soll vorläufig genügen; die Schaltungsbesprechung der Busspeicherkarten heben wir uns für den dritten und letzten Teil dieses Kapitels auf. Wenden wir und nun wieder der Praxis zu: Schon vorhandene und neue Platinen müssen ergänzt oder neu bestückt werden. Die Kontaktleisten, die Verdrahtung und das (neue?) Gehäuse sind anschließend an der Reihe. Hilfestellung bei der eventuell doch notwendigen Fehlersuche wird ebenfalls geboten!

Ausbau des Junior-Computers

Von der Theorie zur Praxis

Nun ist es so weit: Der Lötkolben tritt in Aktion, zumindest, wenn der Ausbau des Junior-Computers beschlossene Sache ist. Anderenfalls dient der nachfolgende Abschnitt dieses Kapitels erst einmal zur Orientierung. Wahrscheinlich wird dann am Ende doch die Erkenntnis stehen, daß die notwendigen Arbeiten nicht allzu schwierig sind und keine unüberwindlichen Hürden genommen werden müssen.
Die meisten Ratschläge und Hinweise zum Bau des Juniors, die Kapitel 1 aus Buch 1 enthält, haben auch hier ihre Gültigkeit. Es kann bestimmt nicht schaden, dieses Kapitel noch einmal sorgfältig zu lesen. Ergänzend dazu hier noch einige allgemeine Anmerkungen:
Das Arbeiten mit einem Lötkolben, dessen Spitze einer Bleistiftmine gleicht, erfordert Lötzinn mit einem Durchmesser von einem halben bis einem Millimeter. Zinnrollen, wie sie von Rohrlegern bei der Montage von kupfernen Wasserrohren verwendet werden, sind hier fehl am Platz! Die Erfahrung lehrt, daß dieser Hinweis keineswegs überflüssig ist.
Die Platine ist vorsichtig und sorgfältig zu behandeln; dies sollte eigentlich selbstverständlich sein. Eine Überhitzung kann die feinen Kupferbahnen ebenso zerstören wie eine scheinbar ungefährliche mechanische Belastung. Das Verschieben der Platine auf einem mit Drahtstücken und Lötzinntropfen übersäten Arbeitstisch ist ebenso gefährlich, wenn dabei auf die Platine gedrückt wird. Vier Schrauben M3 x 40, montiert in den Löchern der Platinenecken, sorgen für sicheren Abstand.
Auch wenn auf IC-Fassungen verzichtet wird, ist mit Sorgfalt darauf zu achten, daß **sämtliche** IC-Pins in ihren Bohrungen stecken und mit den Kupferaugen verlötet werden. Allzu leicht verbiegt sich nämlich ein IC-Pin so, daß er unter dem IC-Gehäuse verschwindet und bei einer Kontrolle alles in Ordnung zu sein scheint. Später ist der umgebogene IC-Pin dann höchstwahrscheinlich die Ursache von unliebsamen und unerklärlichen Überraschungen.
Für alle Drahtverbindungen sollte kein "Klingeldraht", sondern biegsame Litze verwendet werden. Ein Leiterquerschnitt von ca. 0,3 mm reicht für

die meisten Verbindungen aus. Die Leitungen für Masse und +5 V, die zum Netzteil führen, müssen jedoch aus Litze mit einem Querschnitt von mindestens 1 mm^2 bestehen.
Wenn nach der Fertigstellung einer mit ICs bestückten Schaltung der gewünschte Erfolg auf sich warten läßt, investieren viele Hobby-Elektroniker Geld in ein oder mehrere neue ICs. **In 99 von 100 Fällen ist dies vollkommen überflüssig; auch dies lehrt die Erfahrung!** Nur außerordentlich selten ist ein IC bereits defekt, wenn es über den Ladentisch des Einzelhandels wandert. Wesentlich aussichtsreicher als das Wechseln von ICs dürfte fast immer die Kontrolle auf richtige und falsche Verbindungen (kalte Lötstellen und Kurzschlüsse) sein.
Der Arbeitsplan für den Bau der Erweiterungen umfaßt folgende Punkte:

1. Änderungen auf der Basisplatine;
2. Bau der erweiterten und geänderten Speisung;
3. Bestücken der Interfacekarte;
4. Herstellen der Verbindungen zwischen der Basiskarte, der Interfacekarte und, sofern vorgesehen, der Buskarte;
5. Einbau des erweiterten Junior-Computers in ein Gehäuse;
6. Inbetriebnahme und, falls notwendig, Fehlersuche. In dieser Reihenfolge wollen wir nun vorgehen.

Aktionspunkt 1: die Basisplatine

kleinere Widerstände und eine kleine Platine

Um die Basisplatine beliebig drehen und wenden zu können, werden zuerst die Speisespannungsleitungen abgelötet. Tabelle 3 gibt an, was dann zu tun ist: Ein Widerstand muß geändert werden, für drei weitere wird dies eben-

Tabelle 3. Bei der Erweiterung des Junior-Computers auf der Basisplatine durchzuführende Änderungen.

Änderungen auf der Hauptplatine des Junior-Computers:
R5 = 470 Ω (oder 560 Ω parallel zu ursprünglichen R5, unbedingt erforderlich!)
R14,R15,R16 = 470 Ω (oder 560 Ω parallel zu ursprünglichen Widerständen)
1 Drahtbrücke D-EX **(sie ist jedoch nur dann erforderlich, wenn zusätzliche Buskartenspeicher angeschlossen sind;** ansonsten bleibt D mit Masse verbunden).

falls empfohlen. Man kann die vorhandenen Widerstände durch neue ersetzen oder jeweils einen 470 Ω (560 Ω)-Widerstand parallelschalten. Die Parallelwiderstände lassen sich bequem auf die Oberseite der Platine löten, während im anderen Fall die vorhandenen Widerstände entfernt werden müssen. Hierzu kneift man die beiden Anschlußdrähte möglichst dicht am Widerstandskörper ab, erhitzt die Lötstellen mit dem Lötkolben und zieht die Drahtenden mit einer Pinzette heraus. Für "Junior-Junioren" hier der Farbkode der Widerstände:

470 Ω: gelb - violett - braun (-gold)
560 Ω: grün - blau - braun (-gold)
Wer das Systemprogramm PRINTER MONITOR nicht nutzen will oder auf den Step-by-Step-Modus verzichtet (beides wäre bedauerlich!), kann den folgenden Abschnitt überspringen und sofort zum Aktionspunkt 2 übergehen. Jetzt folgen nämlich die Bestückung und der Einbau der **Aufsteckplatine:**

- **Schaltung → Bild 8b**
- **Platine → Bild 11**
- **Bestückung → Tabelle 4**
- **Bauzeichnung → Bild 12**

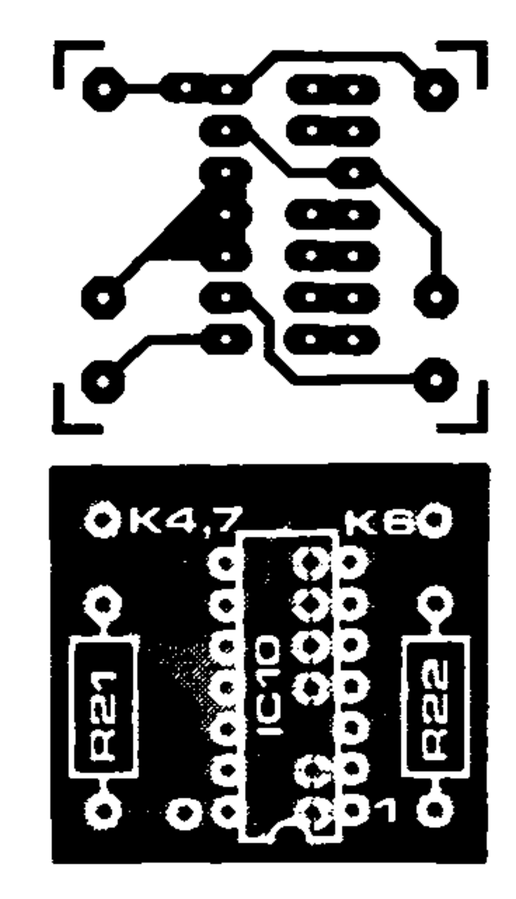


Bild 11. Die Schaltung aus Bild 8b wird auf diese Platine gesetzt. Die kleinen durchkreuzten Kreisringe markieren die Punkte, an die (auf der Kupferseite) kurze, steife Drahtstücke angelötet werden. Diese "Pins" werden in die alte Fassung von IC10 gesteckt.

Zwei bisher nicht benutzte Gatter von IC10 treten ihren Dienst an; sie tragen die Bezeichnungen N7 und N8. Im gleichen Gehäuse sind auch die Gatter N5 und N6 untergebracht. Das benötigte Signal SYNC liegt nicht auf dem Erweiterungskonnektor, so daß diese Ergänzung nicht auf die Interfacekarte verlagert werden konnte. Deshalb wurde als Lösung eine Aufsteckplatine für die Basiskarte gewählt, die an die bisherige Stelle von IC10 tritt.
Wenn IC10 bereits in einer Fassung saß, ist die Änderung nicht schwierig. Anderenfalls muß zuerst IC10 entfernt und dann eine Fassung eingelötet werden, wobei IC10 der Operation zum Opfer fällt: Alle Pins werden mit dem Seitenschneider am IC-Gehäuse abgekniffen und anschließend einzeln ausgelötet. Für die Aufsteckplatine wird dann ein neues IC benötigt.
Neben IC10 befinden sich auf der Aufsteckplatine noch zwei Widerstände und zwei Lötstifte für die K-Leitungen. Auf der Kupferseite sind ferner die Anschlußpunkte ①, ②, ④ . . . ⑦ und ⑭ vorhanden. Diese Punkte stellen die Verbindungen mit der IC-Fassung auf der Basisplatine her; sie sind im Bestückungsaufdruck durch sieben durchkreuzte Kreisringe gekennzeichnet. Hier werden sieben kurze Drahtstücke (versilberter Schalt-

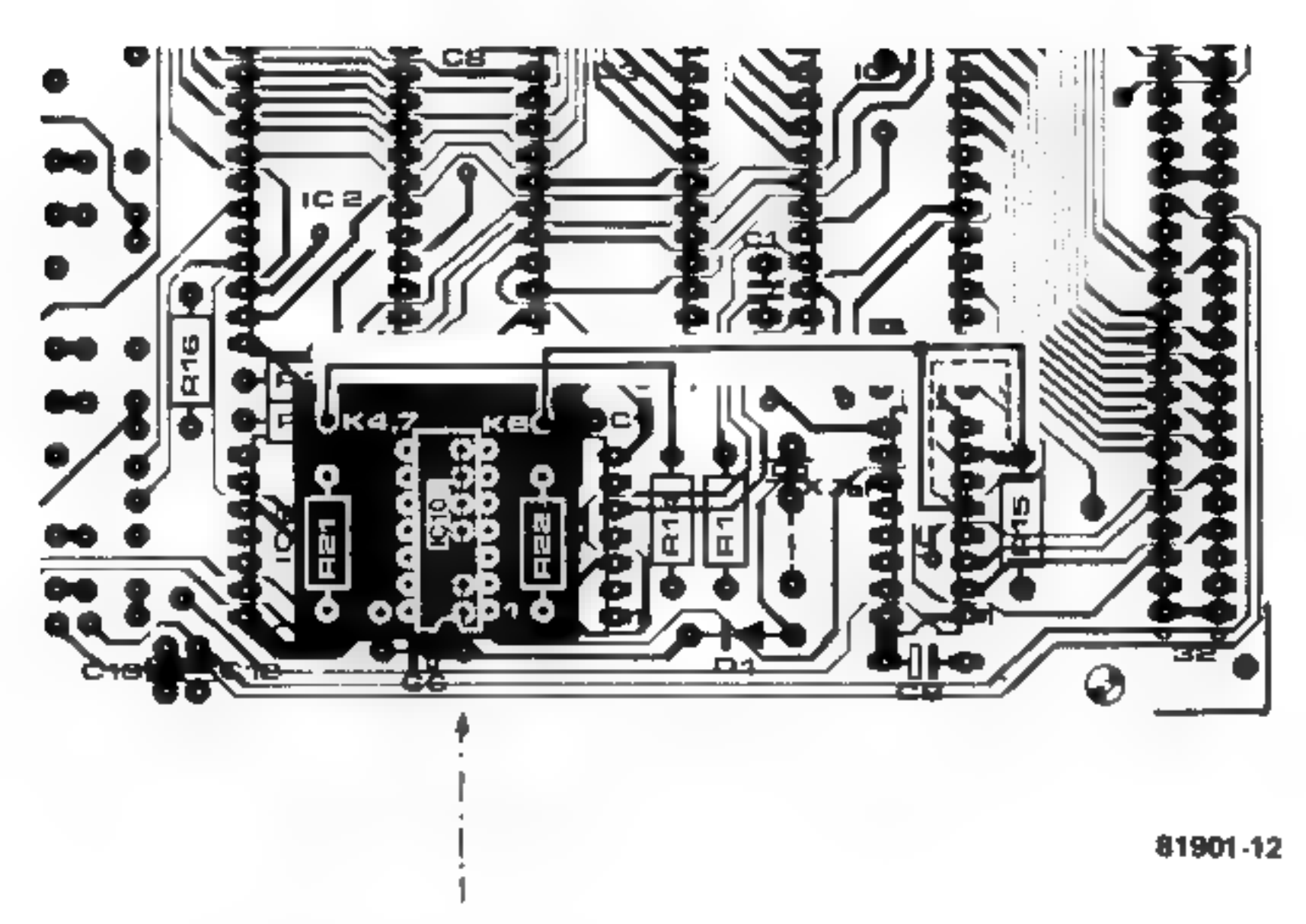


Bild 12. Position der Aufsteckplatine auf der Basisplatine. Wie das Bild zeigt, sind noch zwei Leitungen zu anderen Punkten zu verlegen.

Tabelle 4. Stückliste für die Aufsteckplatine, die an die Stelle von IC10 auf der Basisplatine tritt.

Stückliste Aufsteckplatine (Bild 11)
R21,R22 = 1k (fortlaufende Numerierung in Bezug auf die Hauptplatine des JC)
IC10 = 7401,74LS01 (vorhandenes IC verwenden)
14polige IC-Fassung = DIL-Stecker (siehe Text, evtl. auf Basisplatine vorhanden)
14polige IC-Fassung
2 Lötstifte

draht, ca. 0,3 mm Durchmesser) angelötet. Nachdem die "Pins" auf gleiche Länge abgeschnitten sind, kann die bestückte Aufsteckplatine in die Fassung gesteckt werden.
Natürlich ist auch eine unmittelbare Verbindung zwischen Aufsteckplatine und Basisplatine über sieben Drähte (ohne IC-Fassung) möglich.
Als letztes müssen noch die beiden K-Leitungen verlegt werden. Aus der Bauzeichnung (Bild 12) geht hervor, zu welchen Punkten diese Leitungen auf der Basisplatine führen. Gezeichnet ist der Fall, daß das Systemprogramm PRINTER MONITOR benutzt wird. Will man ohne PM problemlos

dezimal rechnen (siehe Anhang 2), so führt die eine Leitung nicht zu K4, sondern zu K7. Das schrittweise Abarbeiten von Programmen, die im PIA-RAM stehen, ist dann ebenfalls blockiert. Die beiden für die Aufsteckplatine benötigten K-Signale können auch an den betreffenden Lötpunkten des Erweiterungskonnektors abgenommen werden. Hier ist beim Anlöten der Leitungen besonders sorgfältig auf die Vermeidung von Kurzschlüssen zu achten!

Aktionspunkt 2: Das Netzteil

mehr (Milli)Ampere und minus 12 V

- **Schaltung** → **Bild 10**
- **Platine –12 V** → **Bild 13**
- **vorhandene Netzteilplatine** → **Junior-Computer-Buch 1, S. 33 . . . 35**
- **Bestückung** → **Tabelle 5**
- **Bauzeichnung** → **Bild 14**

Beginnen wir mit der vorhandenen Netzteilplatine. Die Dioden D1 und D2 werden entfernt; sie können später als D7 und D8 ihren Dienst tun. Außerdem sind IC1, IC2, IC3 und der Fingerkühlkörper zu demontieren.

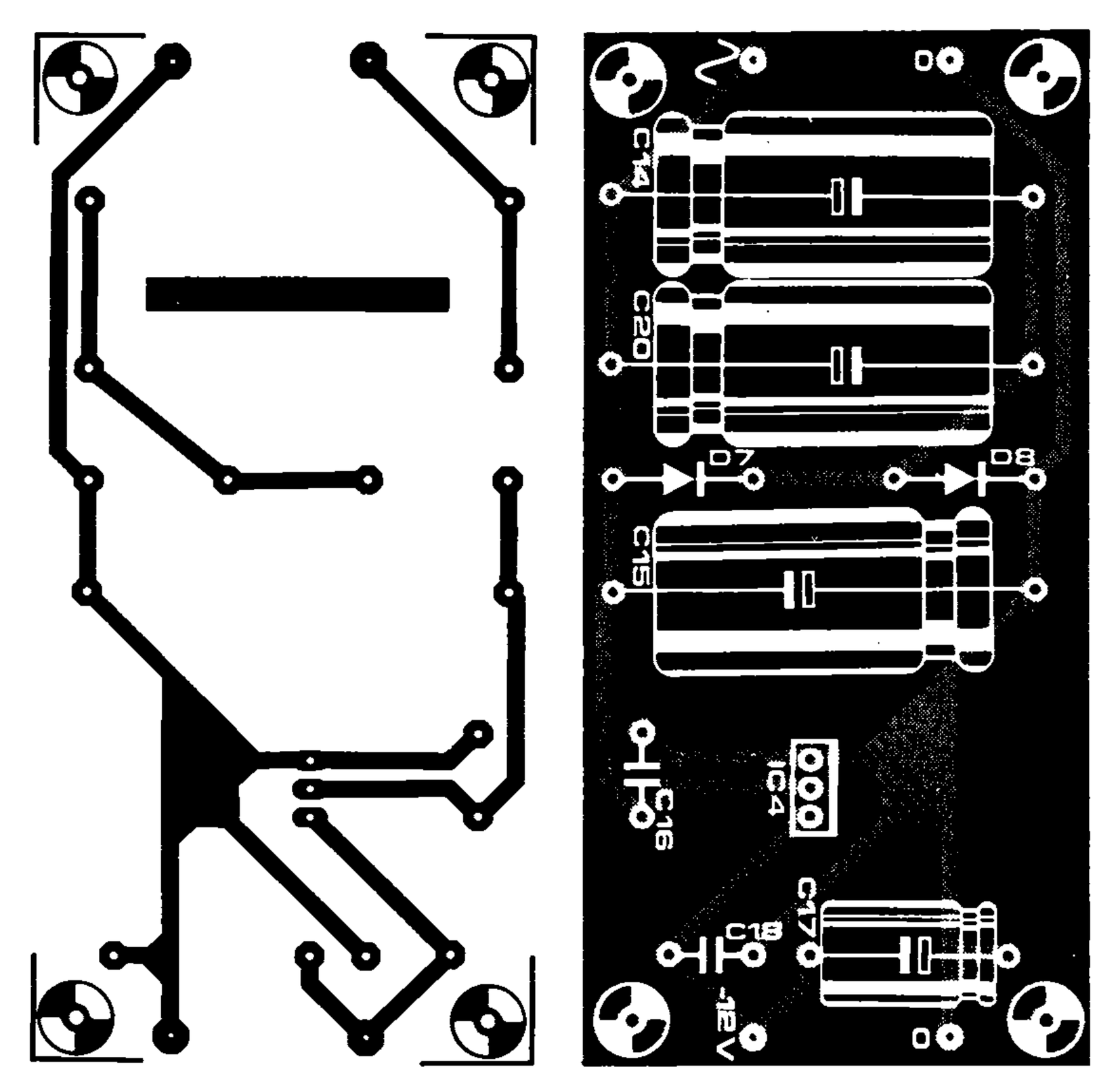


Bild 13. Platine für die –12 V-Speisung; diese Spannung wird für die RS-232-Schnittstelle benötigt.

Auf die Demontage folgt die Neukonstruktion: C19 wird parallel zu C1 geschaltet und, sozusagen im Huckepackverfahren, damit verlötet. Auf richtige Polarität achten! Man kann C1 auch durch einen 680-μ/40-V-Elko ersetzen. In gleicher Weise wird mit C21 bei C6 verfahren, oder C6 wird gegen einen 4700 μ/25 V-Elko ausgetauscht.
Dann folgt die Montage der neuen ICs 1 und 3. Sie werden auf Kühlkörper entsprechend Bild 14b montiert und so eingebaut, daß die IC-Metallflächen in Richtung C2 zeigen. Am besten setzt man die ICs zunächst provisorisch auf die Kühlkörper (IC1 innerhalb der Kühlrippen, siehe Bild 14a) und lötet sie dann so ein, daß die Kühlkörper gerade über den angrenzenden Kondensatoren "schweben". Dabei ist besonders darauf zu achten, daß sich die beiden Kühlkörper nicht berühren. Ist die richtige Position auf der Platine gefunden, dann werden die Kühlflächen der ICs und die Kühlkörper mit Wärmeleitpaste bestrichen und endgültig montiert.
Das neue IC2 hat äußerlich die gleiche Gestalt wie das alte, auch die Anschlüsse sind identisch. Da infolge der erhöhten Stromaufnahme des erweiterten Junior-Computers mehr Wärme entsteht, muß IC2 auf einen größeren Kühlkörper (siehe Bild 14a) gesetzt werden. Der Kühlkörper kann zum Beispiel an die Rückwand des Gehäuses geschraubt werden, wobei die Rippen senkrecht stehen müssen. Damit die Luft ungehindert zirkuliert, darf die Unterkante des Kühlkörpers nicht auf dem Gehäuseboden aufliegen. Einige Löcher im Gehäuseboden unter dem Kühlkörper und an der Oberkante der Rückwand erleichtern die Wärmeabfuhr erheblich. Das Gehäuse muß dazu allerdings auf Füßen stehen. IC2 kann ohne Isolierscheibe auf den Kühlkörper geschraubt werden, da sein Gehäuse gleichzeitig der Masseanschluß ist. Damit liegt dann aber auch das (metallene) Gehäuse an Masse, so daß leitfähige Anstandsröllchen für die Platinen eventuell Kurzschlüsse mit benachbarten Leiterbahnen verursachen können.
Die Bestückung der zusätzlichen −12 V-Netzteilplatine dürfte nach der Feinarbeit auf der Basisplatine keine Schwierigkeiten bereiten. Bild 13 zeigt diese Platine und die zugehörige Bestückung. IC4 wird auf einen Kühlkörper montiert, der vom gleichen Typ wie der Kühlkörper von IC1 und IC3 ist.
Die Bauzeichnung in Bild 14a leistet Hilfestellung bei der Endmontage. Aus diesem Bild geht auch die Verdrahtung des neuen Brückengleichrichters B1 hervor, der ohne Isolation auf den Gehäuseboden oder die -rückwand geschraubt werden kann. Auf die Anschlußzungen des Gleichrichters passen die handelsüblichen Flachstecker, wie sie auch im Bordnetz von Kraftfahrzeugen verwendet werden.
Zur Verdrahtung der einzelnen Baugruppen vorab folgender Hinweis: Die Speisespannungen werden dem erweiterten Junior-Computer über fünf Leitungen (einschließlich der Masseleitung) zugeführt; sie enden auf Lötstützpunkten, die sich auf der Interfacekarte befinden. Der Querschnitt der Leitungen muß ausreichend bemessen sein (Masse- und +5 V-Leitung mindestens 1 mm^2). Empfehlenswert ist ferner eine verschiedenfarbige Isolation der einzelnen Leitungen.
Man vergesse nicht, eine stärkere Sicherung in den Sicherungshalter einzusetzen, da die bisherige Sicherung beim Einschalten des erweiterten Netzteils sofort durchschmelzen dürfte. Diesen Fehler zu finden, kostet manchmal seine Zeit . . .

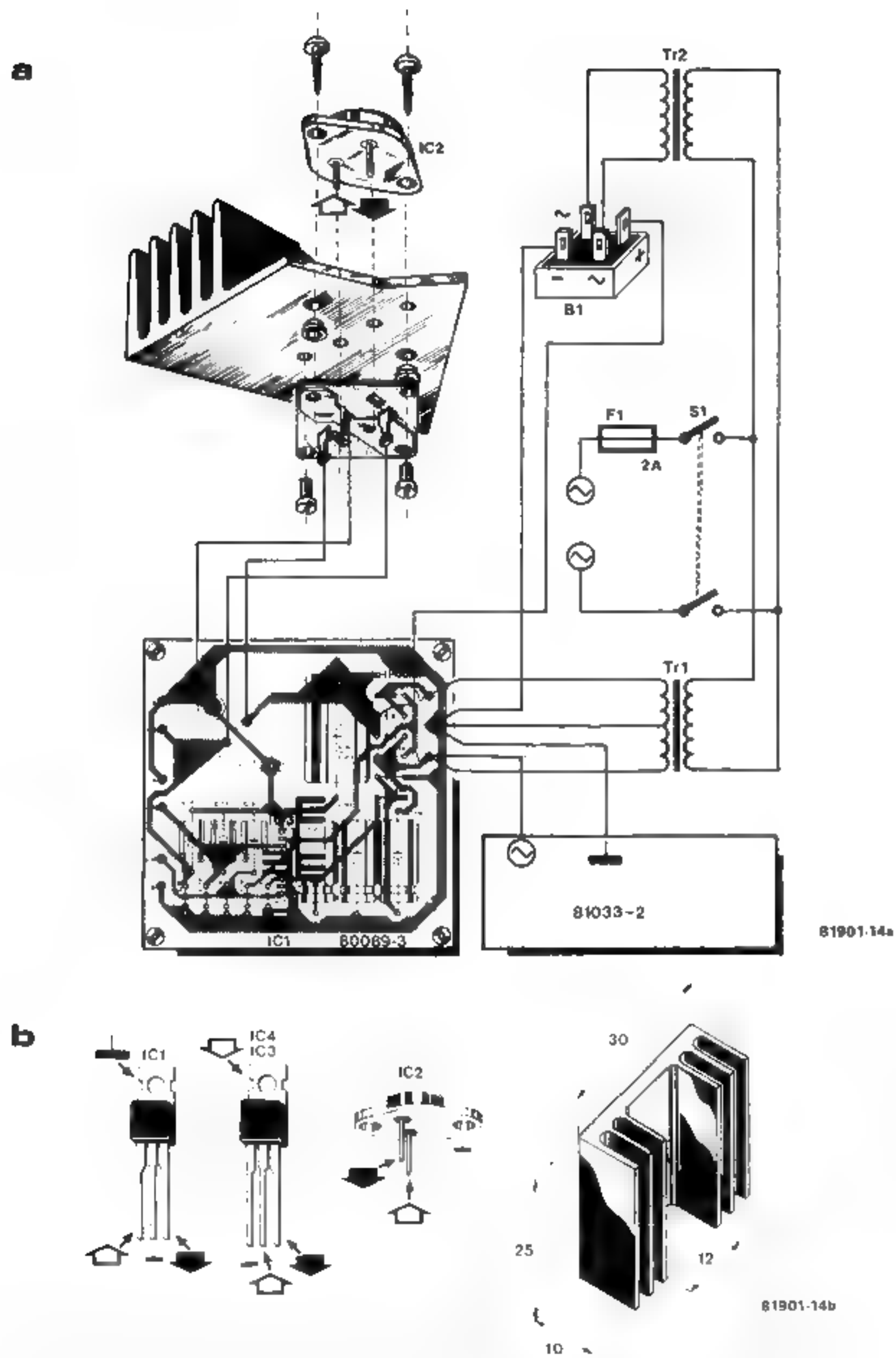


Bild 14. Verdrahtungs- und Bauzeichnung des erweiterten Netzteils (a). Gegenüber der Basisversion sind einige Bauelemente sowie die −12 V-Platine hinzugekommen. Stabilisator IC1 wird mit seinem Kühlkörper extern montiert. Auch Gleichrichter B1 sitzt nicht mehr auf der Platine. Ferner ist die Anschlußbelegung der Spannungsregler und eine Maßzeichnung der Kühlkörper für IC1 und IC3 hinzugefügt (b).

Tabelle 5. Stückliste für die erweiterte Speisung.

Stückliste Netzteil-Platine

Achtung: Gegenüber der vorhandenen Platine ergeben sich einige Änderungen. Die geänderten Bauteile sind in der Stückliste gekennzeichnet.
"%": Das Bauteil ist gegenüber der ursprünglichen Version geändert.
"&": Neue, fortlaufende Numerierung

Kondensatoren:

C1,C2,C10,C14(&),C15(&) = 470 µ/25 V
C3,C11,C17(&) = 47 µ/16 . . . 25 V
C4,C5,C8,C9,C12,C13,C16(&),
C18(&) = 100 n MKH
C6,C21(&) = 2200 µ/25 V (C21//C6)
(oder C6(%) = 4700 µ/25 V; C21 (entfällt)
C7 = 100 µ/25 V
C19(&),C20(&) = 220 µ/40 V
(C19//C1;C20//C14) (C19 und C20 entfallen bei C1 (%), C14 (%) = 680 µ/40 V)

Halbleiter:

IC1(%) = 7812 (TO-220)
IC2(%) = 78H05 (TO-3)
IC3(%) = 7905 (TO-220)
IC4(&) = 7912 (TO-220)
D1,D2 = entfallen; siehe D7 und D8
D3,D4,D5,D6,D7(&),D8(&) = 1N4004
B1(&) = FB 1003

Optische Anzeige:

D9 = LED grün; R1 = 470 Ω: grün-violett-braun-(gold)
D10 = LED rot; R2 = 680 Ω: blau-grau-braun-(gold)

Außerdem:

Tr1 = vorhandener Trafo
Tr2(&) = 1 x 10 V/4 A
S1 = vorhandener Netzschalter
F1(%) = Feinsicherung 2A
Kühlkörper(&) für IC1 . . . IC4

Ganz nützlich, aber nicht unbedingt erforderlich, sind die Kontroll-LEDs D9 und D10. Die rote LED D10 zeigt an, daß Trafo Tr2 (+5 V-Speisung) in Betrieb ist, während die grüne LED D9 für Trafo Tr1 (übrige Spannungen) zuständig ist. Die LEDs werden mit Hilfe von Fassungen auf die Bedienplatte neben den Netzschalter montiert. Nicht auf der Platine befinden sich die Strombegrenzungswiderstände R1 und R2; sie werden unmittelbar an die LEDs angelötet. Für die Identifizierung der Anschlüsse von neuen LEDs hat sich als Gedächtnisstütze die Merkregel K = K bewährt, die besagt, daß das kurze Ende der Katodenanschluß ist. Das Aufleuchten der LEDs D9 und D10 gibt natürlich keinen Aufschluß darüber, ob die Ausgangsspannungen des Netzteils den vorgeschriebenen Wert haben.

Nach Fertigstellung des Netzteils sollte man sich nicht davon abhalten lassen, dieses auf richtige Funktion zu überprüfen. Zuerst werden die Spannungen im Leerlauf und dann unter Vollast gemessen. Als Last für die Spannungen ± 12 V dienen zwei Widerstände 30 Ω/5 W, während –5 V mit 12,5 Ω/2 W und +5 V mit 1,25 Ω/20 W belastet werden. Erlaubt ist bei den Spannungen eine Toleranz von ± 5%. Nun läßt sich auch kontrollieren, ob die Wirkung der Kühlkörper ausreichend ist. Einige Trafo-Ausführungen liefern nämlich zum Ausgleich der Trafoverluste etwas höhere Sekundärspannungen. Diese Verluste sind aber oft geringer als veranschlagt, so daß die Stabilisator-ICs höhere Eingangsspannungen erhalten und folglich mehr Wärme entwickeln. Der Belastungstest ist eigentlich erst sinnvoll, wenn die Kühlkörper in das Gehäuse eingebaut sind und unter Betriebsbedingungen gemessen werden kann.

Aktionspunkt 3: die Interfacekarte

noch mehr Feinarbeit

- **Schaltungen** → **Bild 1 und 2**
- **Bestückung** → **Tabelle 6**
- **Platine** → **Bild 15 (Bestückungsaufdruck)**
  → **Bild 16 (Layout Oberseite)**
  → **Bild 17 (Layout Unterseite)**
  (die Platine ist in verkleinertem Maßstab wiedergegeben)
- **Drahtbrücke IC4/IC5** → **Tabelle 7**
- **IC-Anschlüsse** → **Bild 18**
- **Kontaktleisten** → **Bild 19**

Die Interfaceplatine ist ebenso wie die Basisplatine doppelseitig mit Kupferbahnen bedruckt und durchkontaktiert. Die meisten Bauelemente werden von der gleichen Seite aus eingelötet. Da auf dieser Seite auch die Bestückung aufgedruckt ist, wird sie nachfolgend "Aufdruckseite" genannt. Im Gegensatz zur Basisplatine wurde auf einen doppelseitigen Bestückungsaufdruck verzichtet. Nach dem endgültigen Zusammenbau des erweiterten Junior-Computers zeigt die Aufdruckseite nach unten, so daß die Bezeichnungen "Oberseite" und "Unterseite" an dieser Stelle doppeldeutig wären. Der Bestückungsaufdruck in Bild 15 gibt übrigens die Leiterbahnen wieder, die sich auf der *anderen* Seite der Platine befinden; sie zeigen daher einen spiegelbildlichen Verlauf zu dem Kupferlayout in Bild 17.
Hier die einzelnen Schritte, die aus der leeren Interfaceplatine eine bestückte Interfacekarte werden lassen:

1. Zuerst werden die **Widerstände** eingelötet. Die abgekniffenen Drahtenden können später noch Verwendung als Drahtbrücken finden. Der Farbkode der Widerstände ist in Tabelle 6 angegeben. Abhängig davon, ob R37 entfällt, sind insgesamt 36 oder 37 Widerstände vorhanden. R37 wird nur benötigt, wenn beim INPUT-Kassettenrekorder der Lautsprecher- oder Kopfhörerausgang benutzt werden muß, wobei der eingebaute Lautsprecher automatisch abgeschaltet wird. Einfachere Kassettenrekorder besitzen meistens nur diesen Ausgang. Ist am Rekorder ein Ausgang zum Anschluß eines Verstärkers vorhanden ("line"), so entfällt R37. In diesem Fall wird das Signal von R37 unverhältnismäßig stark abgeschwächt, so daß der PLL

Tabelle 6. Stückliste für die Interfacekarte.

Stückliste Interface-Platine

Widerstände:
R1 . . . R4,R32 . . . R35 = 1 k: braun-schwarz-rot-(gold)
R5 = 22 k: rot-rot-orange-(gold)
R6,R10,R11,R14,R15,
R24,R26 . . . R28 = 10 k: braun-schwarz-orange-(gold)
R7,R8,R36 = 8k2: grau-rot-rot-(gold)
R9,R18,R22,R23 = 4k7: gelb-violett-rot-(gold)
R12 = 6k8: blau-grau-rot-(gold)
R13,R25,R31 = 2k2: rot-rot-rot-(gold)
R16 = 100 Ω: braun-schwarz-braun-(gold)
R17 = 330 Ω: orange-orange-braun-(gold)
R19 = 470 Ω: gelb-violett-braun-(gold)
R20 = 1k2: braun-rot-rot-(gold)
R21 = 15 k: braun-grün-orange-(gold)
R29 = 33 k: orange-orange-orange-(gold)
R30 = 4M7: gelb-violett-grün-(gold)
R37 = 33 Ω: orange-orange-schwarz-(gold)
P1 = 5-k-10 Gang-Trimmpoti
P2 = 1-k-Trimmpoti

Kondensatoren:
C1 = 220 n
C2,C11 . . . C13 = 10 μ/16 V Tantal
C3 = 22 n
C4 = 1 n
C5 . . . C7 = 6n8
C8 = 100 n
C9 = 47 n
C10,C14 . . . C22 = 1 μ/16 V

Halbleiter:
D1 . . , D3 = 1N4148
D4 = LED grün
D5 = LED rot
T1 = BC 547B o.ä.
T2,T3 = BC 516
IC1 = 6522 (Rockwell, Synertek)
IC2,IC3 = 2114
IC4 = 2716 (TM; ESS 506), 2708, 8114
IC5 = 2716 (PM; ESS 507), 2708, 8114
IC6 = NE 565
IC7,IC8 = LM 311
IC9,IC10 = 74LS241
IC11,IC12 = 74LS243
IC13 = 74LS27, 7427
IC14 = 74LS01, 7401
IC15 = 74LS30, 7430
IC16 = 74LS00, 7400
IC17 = 82S23, Signetics, (ESS 508), 74188 (nicht für ESS 508!!)

außerdem:
Re1,Re2 = Reedrelais 1301, 680 Ω (Günther)
J1 . . . J4 = Cinch-Buchse
1 x 25polige Federleiste für RS232 (siehe Bild 19e)
Typ SBM-383 (Siemens)
Best.-Nr.: C42334-A383-A332
1 x Messerleiste 64polig nach DIN 41612 (Siemens) (siehe Bild 19a)
Best.-Nr.: C42334-A191-A502
2 x 8polige IC-Fassung
9 x 14polige IC-Fassung
1 x 16polige IC-Fassung
2 x 18polige IC-Fassung
2 x 20polige IC-Fassung
2 x 24polige IC-Fassung (siehe Text)
1 x 40polige IC-Fassung
20 Lötstifte für VIA-"Konnektor"
29 Lötstifte für gekennzeichnete Anschlüsse
5 Lötstifte für die Verbindungen mit dem Port-Konnektor
5 Lötstifte für die Versorgungsspannung
3 Lötstifte für die LED-Anschlüsse
1 x Ausgangskonnektor 64polig, gebogen, weiblich (siehe Bild 19c und Text)

unter Umständen fehlerhaft arbeitet.

2. Nun folgen die **Trimmpotis.** P2 ist ein Trimmpoti in üblicher Ausführung, während für P1 ein Wendelpoti auf die Platine gesetzt wird. Dieses Poti dient später zum Einstellen des PLL (wird in Kapitel 11 beschrieben).
3. Als nächstes folgen die **MKH-Kondensatoren.** Ihre Schmalseiten sind elektrisch leitend und dürfen nicht mit Anschlüssen oder Metallteilen anderer Bauelemente in Berührung kommen. Leider ist dieser Hinweis durchaus nicht überflüssig.
4. Die **Tantalelkos.** Hier ist beim Einlöten auf richtige Polarität zu achten. Bei den üblichen Ausführungen ist nur der Plusanschluß gekennzeichnet.
5. **Drahtbrücken und Lötstifte.** Wenn wir richtig gezählt haben, sind auf der Interfacekarte 62 Lötpunkte für den Anschluß von Drahtbrücken vorhanden. Obwohl in der Stückliste (Tabelle 6) Lötösenstifte angegeben sind, können alle Drahtbrücken und Drahtanschlüsse natürlich auch unmit-

telbar eingelötet werden. Die betreffenden Platinenbohrungen haben aus herstellungstechnischen Gründen den gleichen Durchmesser wie die Bohrungen für die Anschlüsse der Bauelemente; **es sind daher nur Lötösenstifte mit dünnem Schaft geeignet.**
Beim Anschluß des Elekterminals und der meisten Drucker entfällt die Drahtbrücke P-Q. Die Lage einer anderen Drahtbrücke ist vom späteren Vorhandensein der Busspeicherkarten abhängig:

Busspeicherkarten vorhanden	→	Drahtbrücke R-S (und D-EX auf der Basiskarte!)
keine Busspeicherkarten	→	Drahtbrücke R-T (und D-⊥ auf der Basiskarte)

Die Lage der übrigen Brücken zwischen den mit Buchstaben bezeichneten Punkten richtet sich nach dem für IC4 und IC5 verwendeten Speicher-IC-Typ. Tabelle 7 gibt darüber genaue Auskunft.
Auch die beiden Drahtbrücken am Ausgangskonnektor (Aufdruckseite) und die drei Brücken am RS-232-Konnektor (Kontakte 4 & 5, 5 & 8 und 6 & 20) dürfen nicht vergessen werden. Die Brücken am RS-232-Konnektor sind beim Anschluß des Elekterminals zwar nicht unbedingt notwendig, doch ist das Einlöten später etwas schwieriger.

6. **IC-Fassungen.** Es spricht nichts dagegen, die ICs und auch die DIL-Relais in Fassungen zu setzen. Diese Bauelemente lassen sich dann leicht auswechseln, wenn es doch einmal notwendig werden sollte. Natürlich müssen nicht nur die richtigen ICs an den richtigen Stellen auf der Platine sitzen, sondern die ICs müssen auch in der richtigen Position in ihrer Fassung stecken. DIL-ICs lassen sich bekanntlich auch um 180° verdreht einsetzen. Die meisten ICs sind deshalb mit einer Aussparung im Gehäuse oder mit einem anderen Zeichen versehen, das den Pin 1 markiert. Auch Fassungen tragen häufig ein solches Zeichen, obwohl ihre Einbaulage beliebig ist. Es empfiehlt sich, solche Fassungen so auf die Platine zu löten, daß beim Einsetzen der ICs keine Irrtümer entstehen.
Die Pins der einzulötenden IC-Fassung werden in die Bohrungen "eingefädelt"; anschließend drückt man die Fassung so auf die Platine, daß sie auf dieser überall aufliegt. Zuerst werden zwei diagonal gegenüberliegende Pins, zum Beispiel die Pins 1 und 8 bei einer 14-Pin-DIL-Fassung, provisorisch angelötet. Nun braucht die Fassung nicht mehr festgehalten zu werden, so daß man beide Hände frei hat, um zuerst die übrigen Pins und dann die provisorisch befestigten Pins endgültig zu verlöten.
Für die EPROMs auf der Interfacekarte (IC4 und IC5) sind eventuell sogenannte Wechselfassungen von Vorteil. Diese Fassungen besitzen eine Klemmvorrichtung, die die Pins des eingesetzten ICs festhält. Nach Lösen der Sperre läßt sich das IC herausnehmen, ohne daß dazu mechanische Kraft notwendig ist. Ein EPROM kann dann leicht gegen ein anderes (mit anderem Inhalt) ausgewechselt werden.

7. **Transistoren und Dioden.** Auch diese Bauelemente müssen lagerichtig eingelötet werden. Bei den Dioden kennzeichnet der breiteste Farbring die Katode; dieser Anschluß entspricht dem Strich des Schaltbildsymbols.

8. **LEDs.** Die LEDs D4 und D5 des Kassetteninterface werden später auf das Bedienpult montiert. Auf der Platine befinden sich nur die Anschlußpunkte. Merkregel für neue LEDs: kurzer Anschluß = Katode!

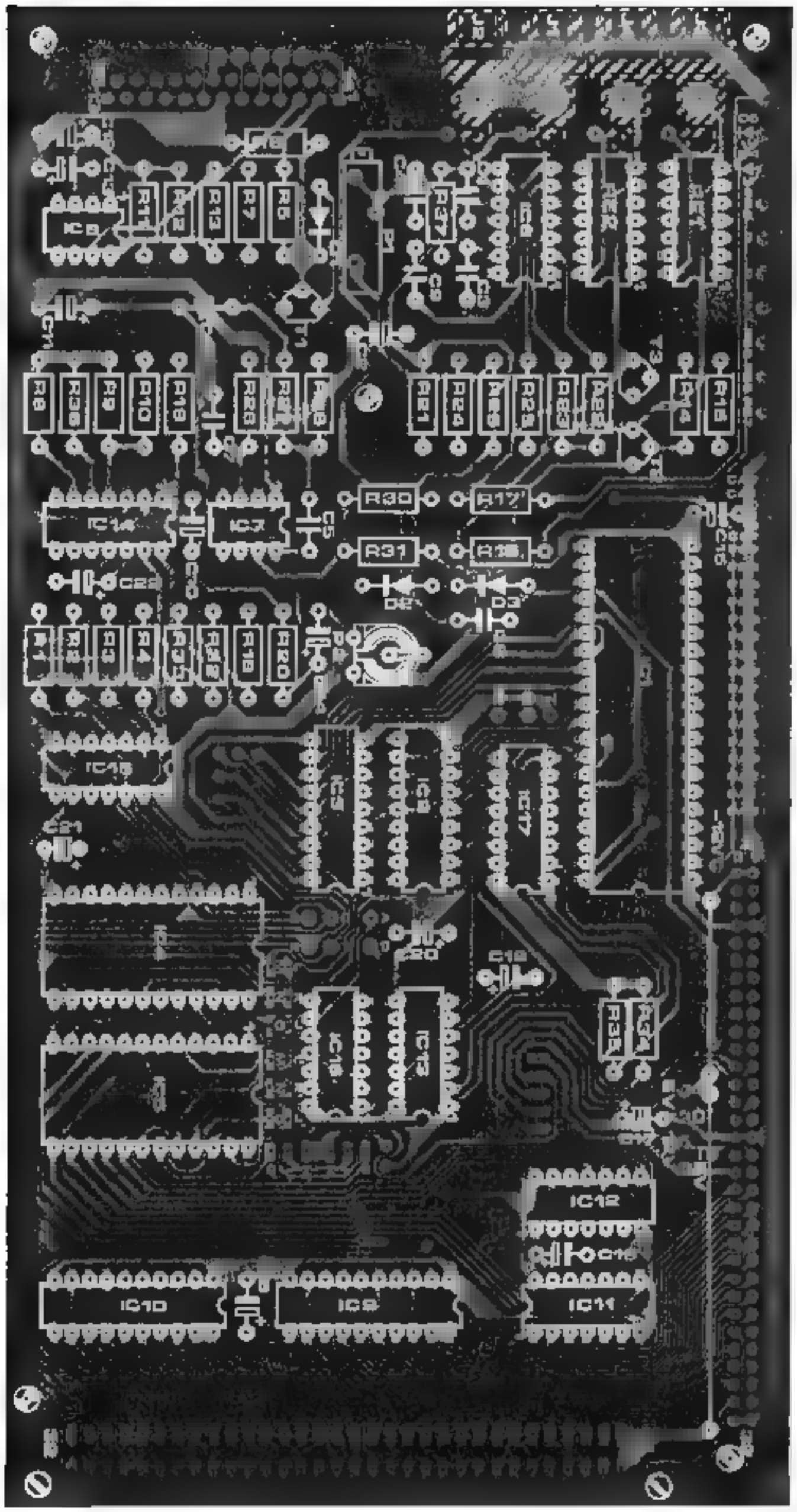


Bild 15. Aufdruckseite der Interfacekarte zusammen mit dem Verlauf der Leiterbahnen auf der Unterseite ("Kupferseite"). Die Platine ist hier verkleinert wiedergegeben!

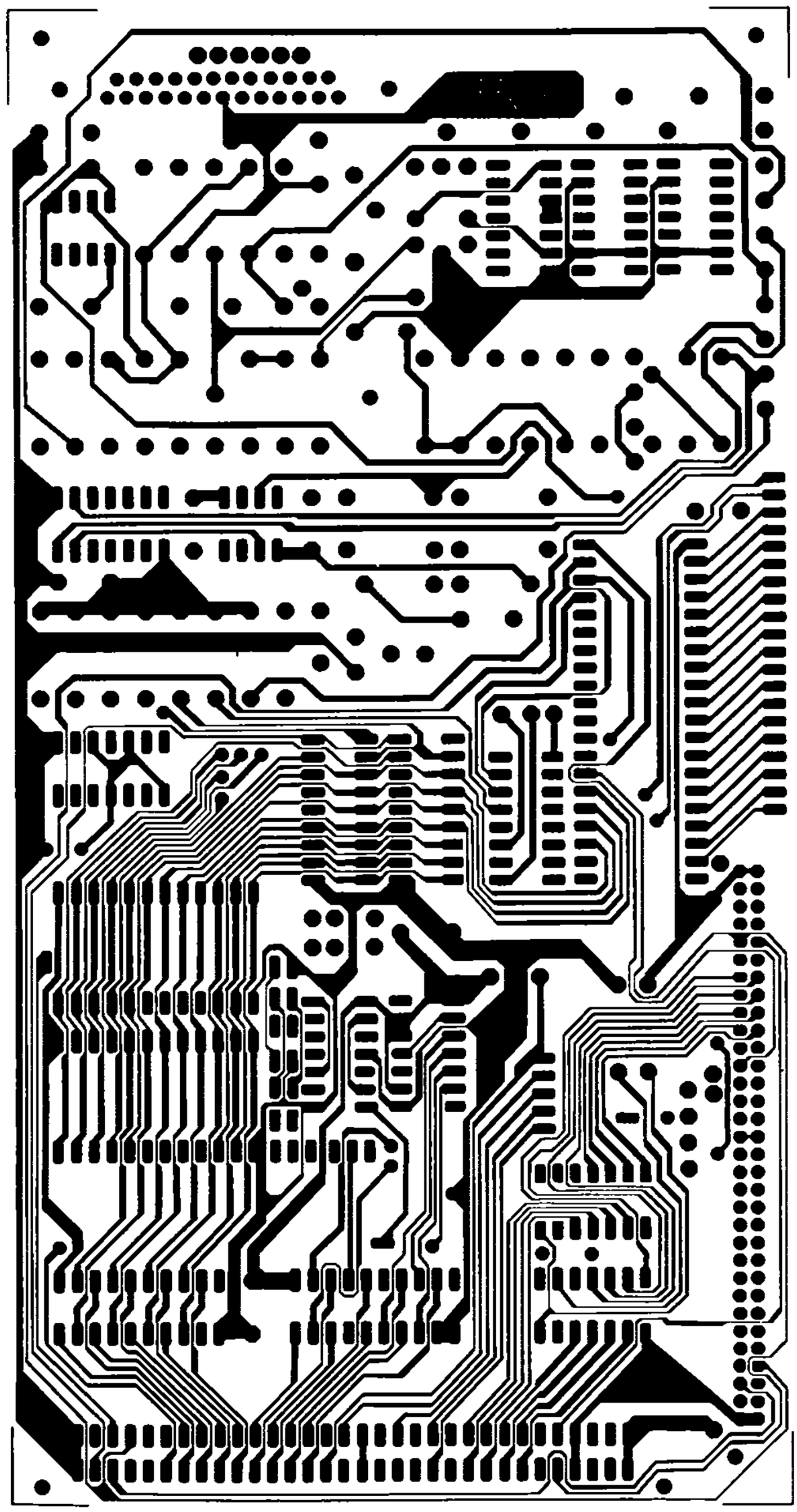


Bild 16. Verlauf der Leiterbahnen auf der Oberseite ("Aufdruckseite") der Interfacekarte. Verkleinerte Wiedergabe!

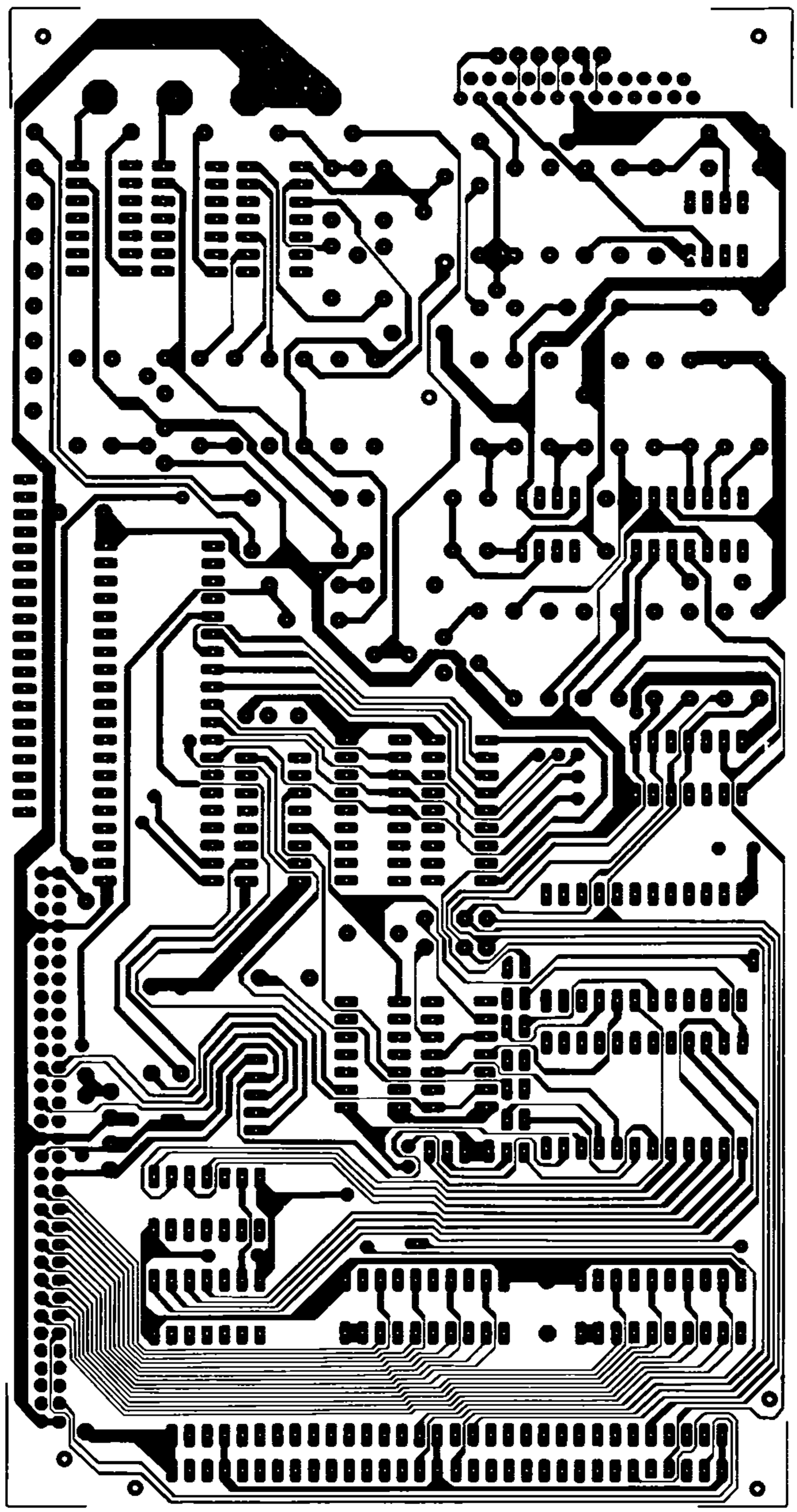


Bild 17. Verlauf der Leiterbahnen auf der Unterseite ("Kupferseite") der Interfacekarte. Verkleinerte Wiedergabe!

Tabelle 7. Lage der Brücken auf der Interfacekarte in Abhängigkeit vom verwendeten EPROM-Typ.

IC	Speicher	Typ	G . . . O G'. . . O'	A . . . F	Speicherbereich
IC4	1K-RAM	8114	O - M	A - B	0800 . . . 0BFF
	1K-EPROM	2708	O - N G - H J - K	A - B	0800 . . . 0BFF
	2K-EPROM	2716	O - N G - I J - L	A - B - C[1]	0800 . . . 0FFF[1]
IC5	1K-RAM	8114	O'- M'	D - C[2] D - E[4]	0C00 . . . 0FFF[2] 1000 . . . 13FF[4]
	1K-EPROM	2708	O'- N' G'- H' J' - K'	D - C[3] D - E[4]	0C00 . . . 0FFF[3] 1000 . . . 13FF[4]
	2K-EPROM	2716	O'- N' G'- I' J' - L'	D - E - F[5]	1000 . . . 17FF[5]

1. bestimmt für Systemprogramm TAPE MONITOR (TM); ESS 506
2. vorzugsweise, wenn IC4 = 8114 (geschlossener RAM-Bereich)
3. vorzugsweise, wenn IC4 = 2708 (geschlossener EPROM-Bereich) oder wenn IC4 = 8114 (geschlossener Speicher-Bereich)
4. bei IC4 = 2716
5. bestimmt für Systemprogramm PRINTER MONITOR (PM); ESS 507. Es sind auch noch weitere K-Anschlüsse möglich. Hier sind nur die relevantesten angegeben.

Selbstverständlich sind noch mehr K-Anschlußmöglichkeiten denkbar. Die Auswahl in der Tabelle ist auf die sinnvollsten Verbindungen begrenzt.

9. **ICs.** Man achte auf richtige Position! Falsch (um 180° verdreht) eingesetzte ICs können beim Einschalten der Speisespannung irreparabel beschädigt werden. Eine vorherige Kontrolle vermeidet Ärger und Kosten! Mindestens ein IC muß programmiert sein: das PROM IC17 (siehe Tabelle 2). Beim Typ 82S23 kann dies der Elektor-Programmierservice übernehmen (näheres siehe aktuelle Ausgabe der Zeitschrift "Elektor"), beim ebenfalls verwendbaren Typ 74188 ist das nicht möglich. Wählt man für IC4 und IC5 EPROMs, dann müssen auch diese programmiert sein. EPROMs des Typs 2716 kann man vom Elektor-Programmierservice mit der Systemsoftware TM und PM "füllen" lassen, oder man programmiert die EPROMs an Hand des Hexdump (siehe am Schluß dieses Buches) selbst. Dafür muß allerdings ein Programmiergerät zur Verfügung stehen. Ohne TM ist sämtliche Kassettenhardware nutzlos, es sei denn, daß alternative Kassettensoftware vorhanden ist. Der Gebrauch von PM setzt die Existenz von TM voraus, da TM auch für die Tastenfunktionen G und S von PM benötigt wird.
10. Die **Relais.** Re1 und Re2 sind nur dann von Nutzen, wenn an den erweiterten Junior-Computer zwei Kassettenrekorder angeschlossen werden: der eine zum Aufzeichnen und der andere zum Laden von Daten.

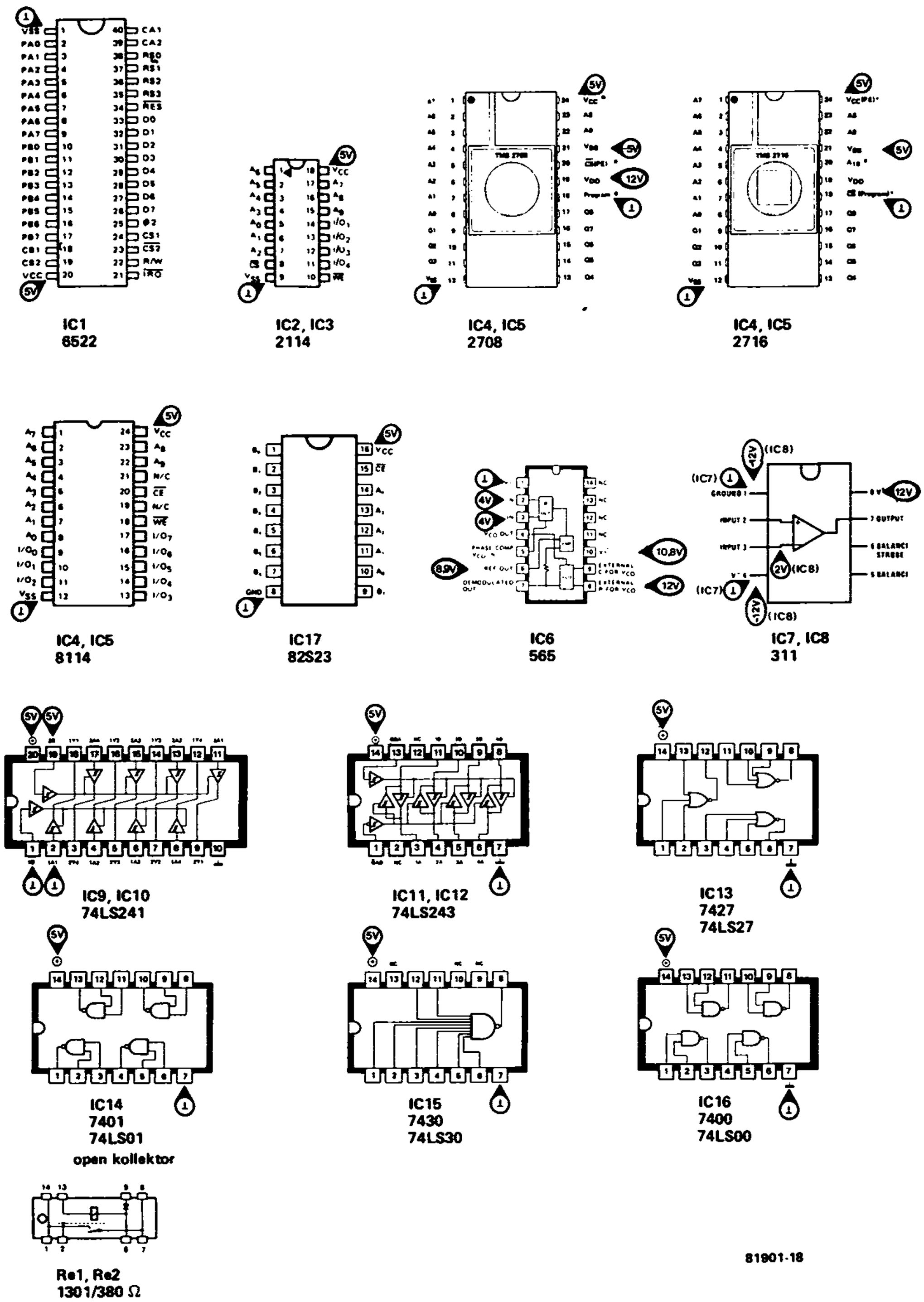
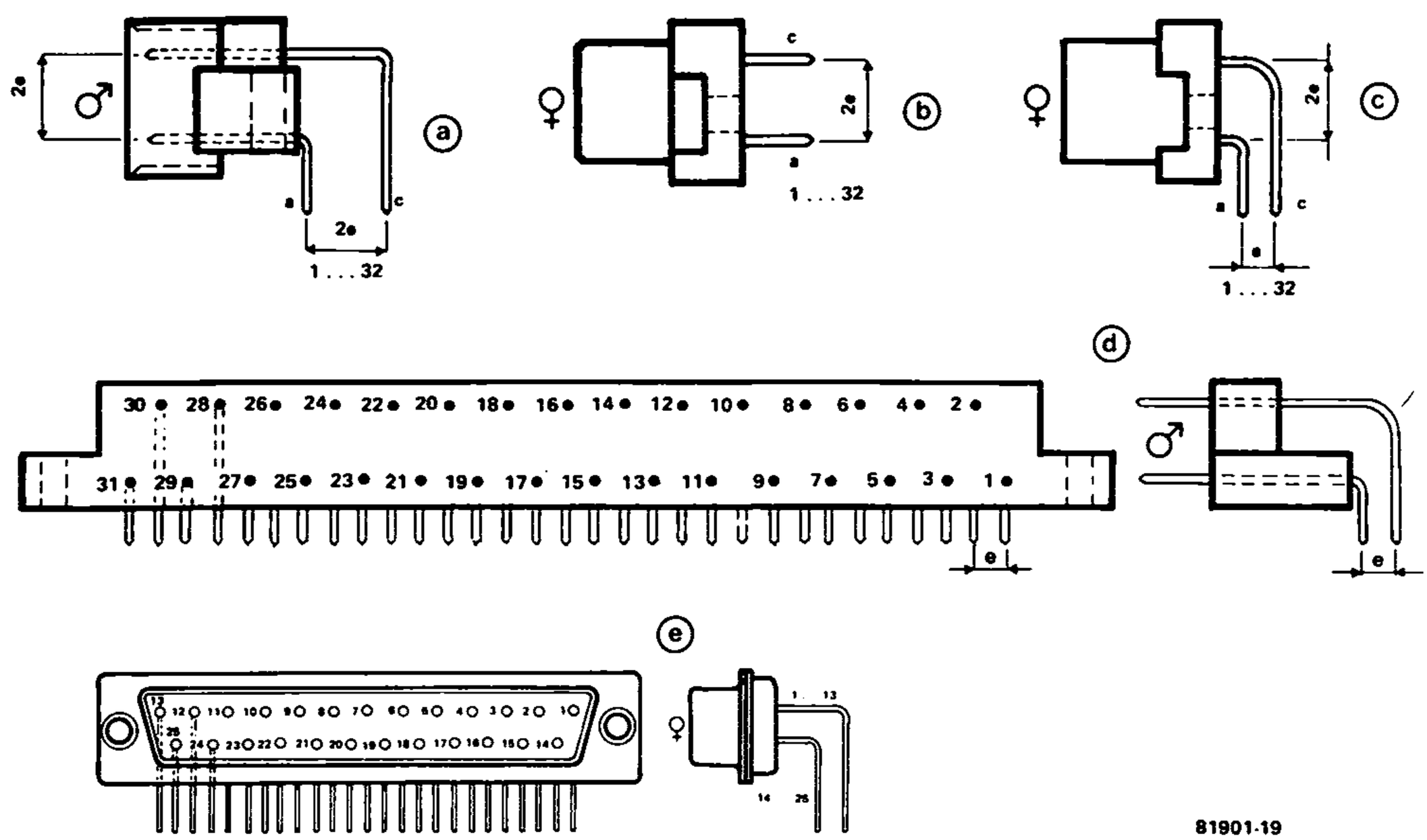


Bild 18. Pinbelegung aller auf der Interfacekarte befindlichen ICs mit Angabe einiger Gleichspannungswerte.

Bild 19. Übersicht über die verschiedenen Konnektortypen, soweit sie für den Junior-Computer von Interesse sind.



Ferner muß an beiden Kassettenrekordern ein Fernbedienungsanschluß (Remote Control) vorhanden sein, oder es muß ein solcher Anschluß eingebaut werden. In allen anderen Fällen können die Relais, die zugehörigen Fassungen und die Buchsen J3 und J4 entfallen.

11. Die **Buchsen J1 . . . J4** für die Verbindungen zu den Kassettenrekordern. Sie können unmittelbar auf die "Kupferseite" der Platine gelötet werden und sind dann später durch Löcher in der rechten Gehäuseseitenwand erreichbar. Eine andere Lösung ist die Montage auf der Rückseite des Gehäuses. J3 und J4 müssen, falls sie überhaupt notwendig sind (siehe Punkt 10), von Masse isoliert montiert werden. J1 und J2 sind über abgeschirmte Leitungen mit der Platine zu verbinden.

12. Der **Eingangskonnektor** wird, falls notwendig (siehe Aktionspunkt 4), auf die dafür bestimmte Schmalseite der Interfacekarte montiert. Der Konnektorkörper sitzt auf der Aufdruckseite der Platine. Hier gelten im wesentlichen die gleichen Montagevorschriften wie bei den IC-Fassungen (Punkt 6). Siehe auch Bild 19a.

13. Der **RS-232-Konnektor.** Wenn der 25-polige D-Konnektor auf die Platine montiert wird, muß die in Tabelle 6 angegebene und in Bild 19e dargestellte stehende Ausführung mit abgewinkelten Lötanschlüssen verwendet werden. Der Körper des Konnektors befindet sich **nicht** auf der Aufdruckseite, sondern auf der "Kupferseite" der Platine. Die Montage auf der falschen Seite führt zu einer gespiegelten und damit falschen Stiftbelegung. Die Lage des RS-232-Konnektors wurde deshalb wie beschrieben gewählt, weil so der Raum zwischen Basis- und Interfacekarte besser

genutzt werden kann. Bei Platinenmontage des Konnektors ist ein Durchbruch in der rechten Gehäuseseitenwand erforderlich, durch den der Konnektor von außen zugänglich ist. Soll der Konnektor an der Rückwand montiert werden, so wird eine für freie Verdrahtung geeignete Ausführung benötigt; sie wird über Flachkabel mit der Interfacekarte verbunden.

14. Für den **VIA-"Konnektor"** sind auf der Platine 20 Lötstifte oder Drahtanschlüsse vorgesehen. Er ist in erster Linie für den Anschluß von Applikationen bestimmt, so daß hierfür zweckmäßigerweise eine Steckverbindung auf die Gehäuserückseite oder auf das Bedienpult montiert wird. Geeignet ist ein 31-poliger Konnektor (weiblich) vom gleichen Typ wie der PIA-Portkonnektor auf der Basisplatine. Für die Verbindungen bietet sich auch hier Flachkabel an.

15. Der **Ausgangskonnektor** bildet das Schlußlicht. Dieser Konnektor entfällt, solange keine Busspeicherkarten auf die Interfacekarte gesteckt werden sollen. Doch auch wenn derartige Karten vorhanden sind, können sie natürlich ebenfalls über Flachkabel mit der Interfacekarte verbunden werden.

Entschließt man sich für eine steckbare Verbindung mit der Buskarte, dann kann man die in Bild 19c gezeichnete Konnektorausführung verwenden (Montage auf der "Kupferseite"). Dabei ist jedoch ein kleines Problem zu lösen: Der Abstand zwischen der Lötpunktreihe für die Stifte a und dem Platinenrand ist nämlich so groß, daß der Konnektor nicht bündig mit dem Platinenrand montiert werden kann. Schuld sind daran einige Leiterbahnen am Platinenrand, für die an anderer Stelle kein Platz ist. Setzt man den Konnektor ohne weitere Maßnahmen auf die Platine, dann sind die Lötanschlüsse für eine einwandfreie Verbindung zu kurz. Um Abhilfe zu schaffen, kann man die Enden so biegen, daß genügend Länge gewonnen wird. Oder man nimmt die in Bild 19b dargestellte Konnektorausführung mit einer Anschlußdrahtlänge von mindestens 13 mm (z.B. eine Wire-Wrap-Version) und biegt die Anschlüsse entsprechend den Erfordernissen um. Die Montage gegen den Rand der Platine ist auf diese Weise möglich. Eine weitere Lösung besteht darin, einen Konnektor zu verwenden, der für freie Verdrahtung vorgesehen ist. Die einzelnen Kontakte werden dann über insgesamt 64 kurze Drähte mit der Platine verbunden. Bei größerem Abstand, zum Beispiel beim Anschluß von Buskartenspeichern über einen Konnektor an der Gehäuserückwand, werden diese Verbindungen über Flachkabel hergestellt.

Siehe hierzu auch die Bilder 20b und 20c; alles weitere geht aus dem nächsten Abschnitt hervor.

Aktionspunkt 4: Verdrahtung

Karten auf den Tisch!

- **Bauzeichnung** → Bild 20 (Details)
- **Stückliste** → Tabelle 8
- **Konnektoren** → Bild 19
- **Buskarte** → Bild 21 (verkleinerter Maßstab)

Das Zusammenfügen der einzelnen Karten zu einem funktionsfähigen und bedienfreundlichen System steht in engem Zusammenhang mit der

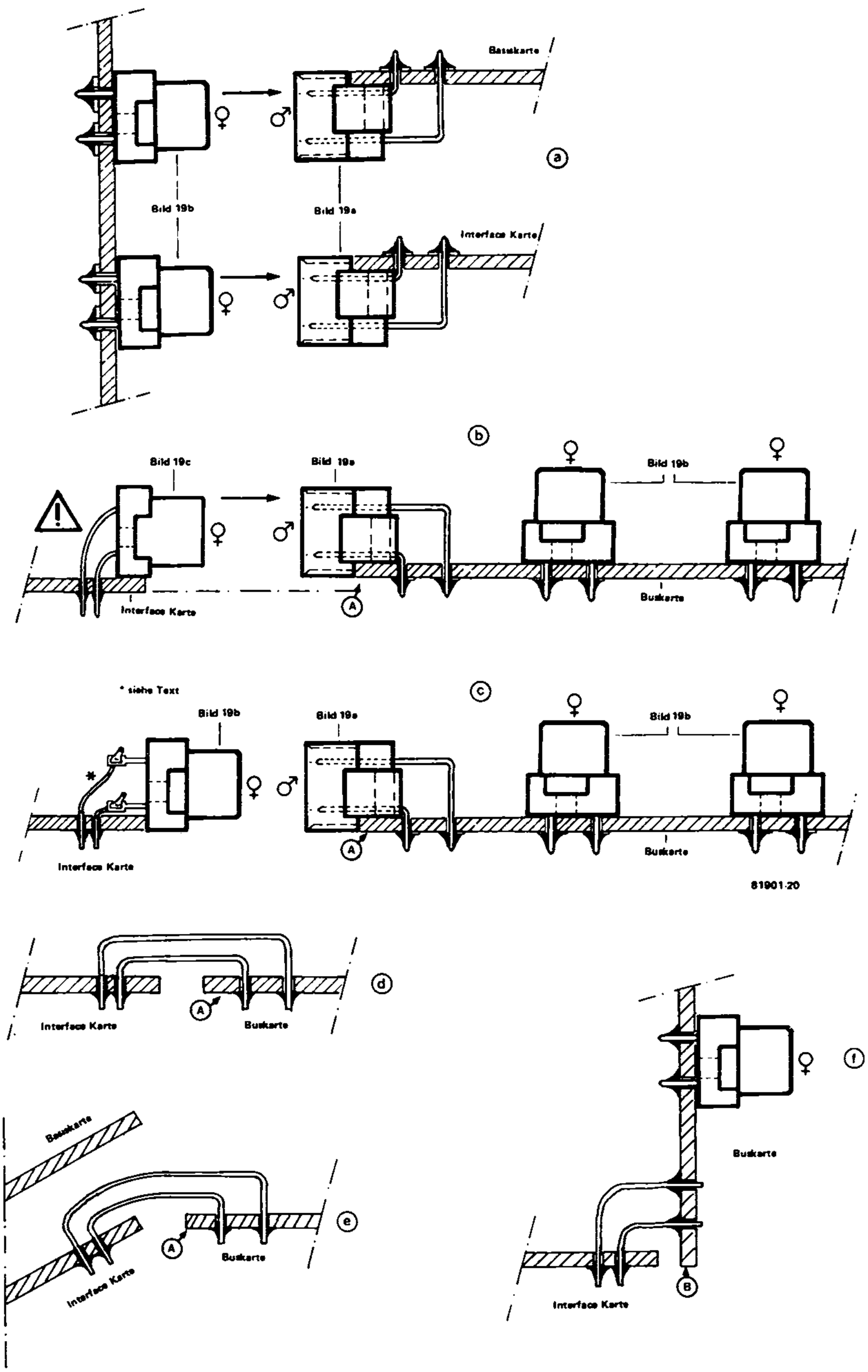


Bild 20. Einige Konstruktionsdetails für die elektrische Verbindung der Karten durch Konnektoren und Drahtverbindungen.

Tabelle 8. Elektrische Verbindungen des Gesamtsystems.

Elektrische Verbindungen im Gesamtsystem
a) Basiskarte ↔ Interfacekarte zwei 64polige Konnektoren (weiblich mit geraden Anschlüssen (Bild 19b) 1 Platine EPS 80024 (davon werden nur drei Konnektoranschlußmöglichkeiten benötigt) oder 1 Lochrasterplatine (Raster 2,54 mm) ein 31poliger Konnektor (männlich) mit abgewinkelten Anschlüssen (Bild 19d)
b) Interfacekarte ↔ Buskarte ein 64poliger Konnektor (männlich) mit abgewinkelten Anschlüssen (Bild 19a) eins bis fünf 64polige Konnektoren (weiblich) mit geraden Anschlüssen (Bild 19b) 1 Platine EPS 80024 oder 1 Lochrasterplatine (Raster 2,54 mm)

Gestaltung des Gehäuses. Während die Konstruktion des Gehäuses unter den nächsten Aktionspunkt fällt, geht es hier zunächst um die Verbindung der einzelnen Karten.
Die Abmessungen der Interfacekarte stimmen mit den Abmessungen der Basiskarte überein; auch die Lage der fünf Befestigungslöcher ist bei beiden Karten gleich. Die Karten können deshalb leicht zu einem platzsparenden "Sandwich" zusammengeschraubt werden. Oben muß natürlich die Basiskarte liegen, damit das hexadezimale Tastenfeld zugänglich ist. Die beiden Platinen werden so montiert, daß ihre "Lötseiten" zueinanderzeigen. Bestimmend für den Abstand zwischen den Platinen ist dann die Höhe der beiden Schalter S24 und S25. Der benötigte Zwischenraum ist so schmal, daß das Bedienpult später nach ergonometrischen Gesichtspunkten optimal gestaltet werden kann. Wir wiederholen hier noch einmal, was bereits unter Aktionspunkt 3 gesagt wurde: Die Interfacekarte bildet, mit der Kupferseite nach oben, die untere Schicht des Sandwich; ihre Aufdruckseite zeigt folglich nach unten. In dieser Lage muß der Eingangskonnektor links auf der nach unten gekehrten Seite und der RS-232-Konnektor (sofern montiert) rechts auf der nach oben zeigenden Seite der Platine liegen. Der Erweiterungskonnektor der Basiskarte und der Eingangskonnektor der Interfacekarte befinden sich also an den linken Schmalseiten ihrer Platinen. Hier müssen 64 einzelne Verbindungen, zum Beispiel mit Hilfe von 64-poligen Konnektoren geschaffen werden.
Dann die Buskarte: Der Ausgangskonnektor, auf den die Buskarte aufgesteckt wird, sitzt auf der vom Bedienenden abgewandten Längsseite der Interfacekarte, und zwar auf der zur Basiskarte gekehrten Platinenseite in unmittelbarer Nähe des Eingangskonnektors. Von hier aus zeigt die Buskarte vom Bedienenden weg in gleiche Richtung wie die Interfacekarte. Die auf die Buskarte gesteckten Speicherkarten ragen dann senkrecht in die Höhe.

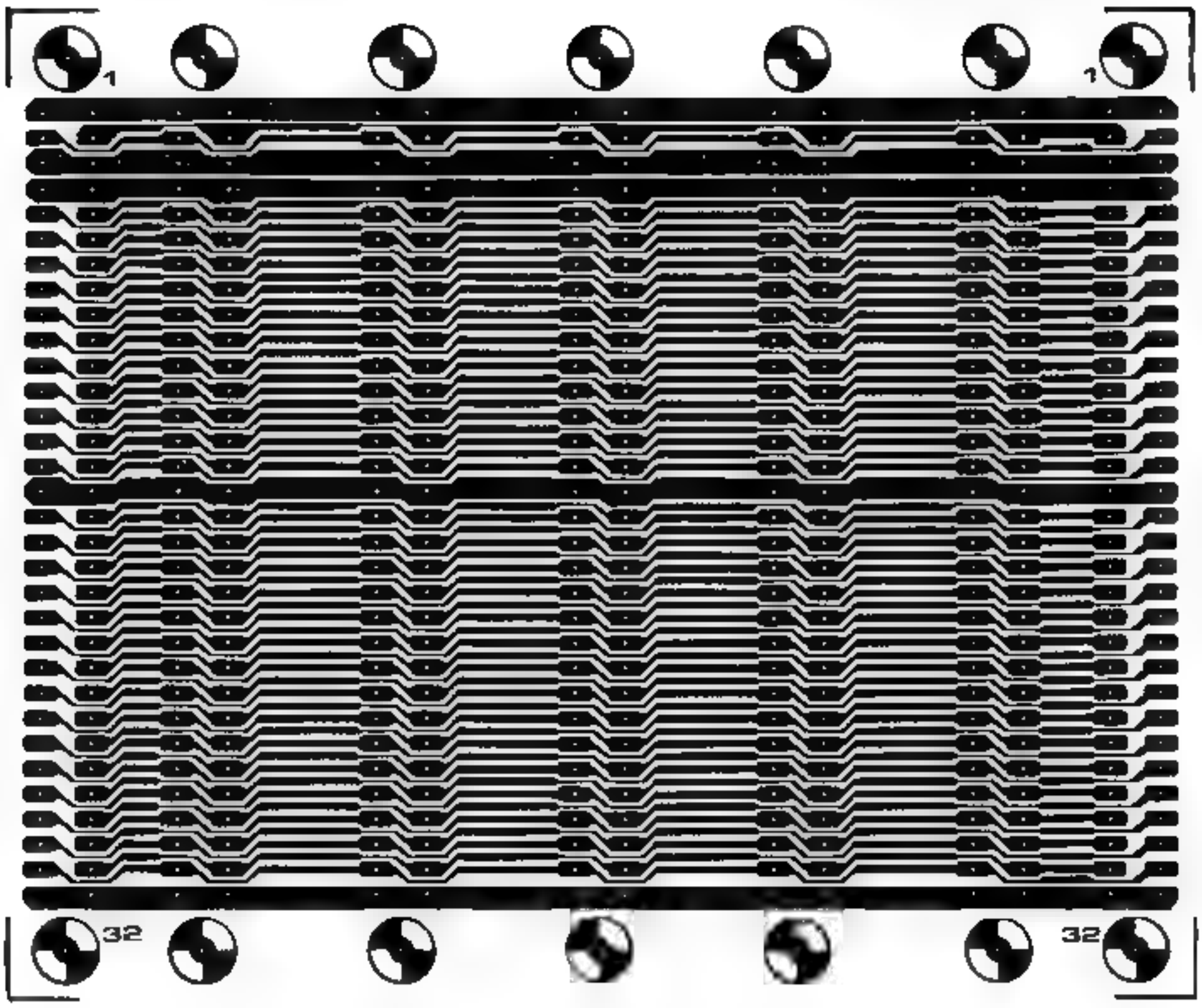


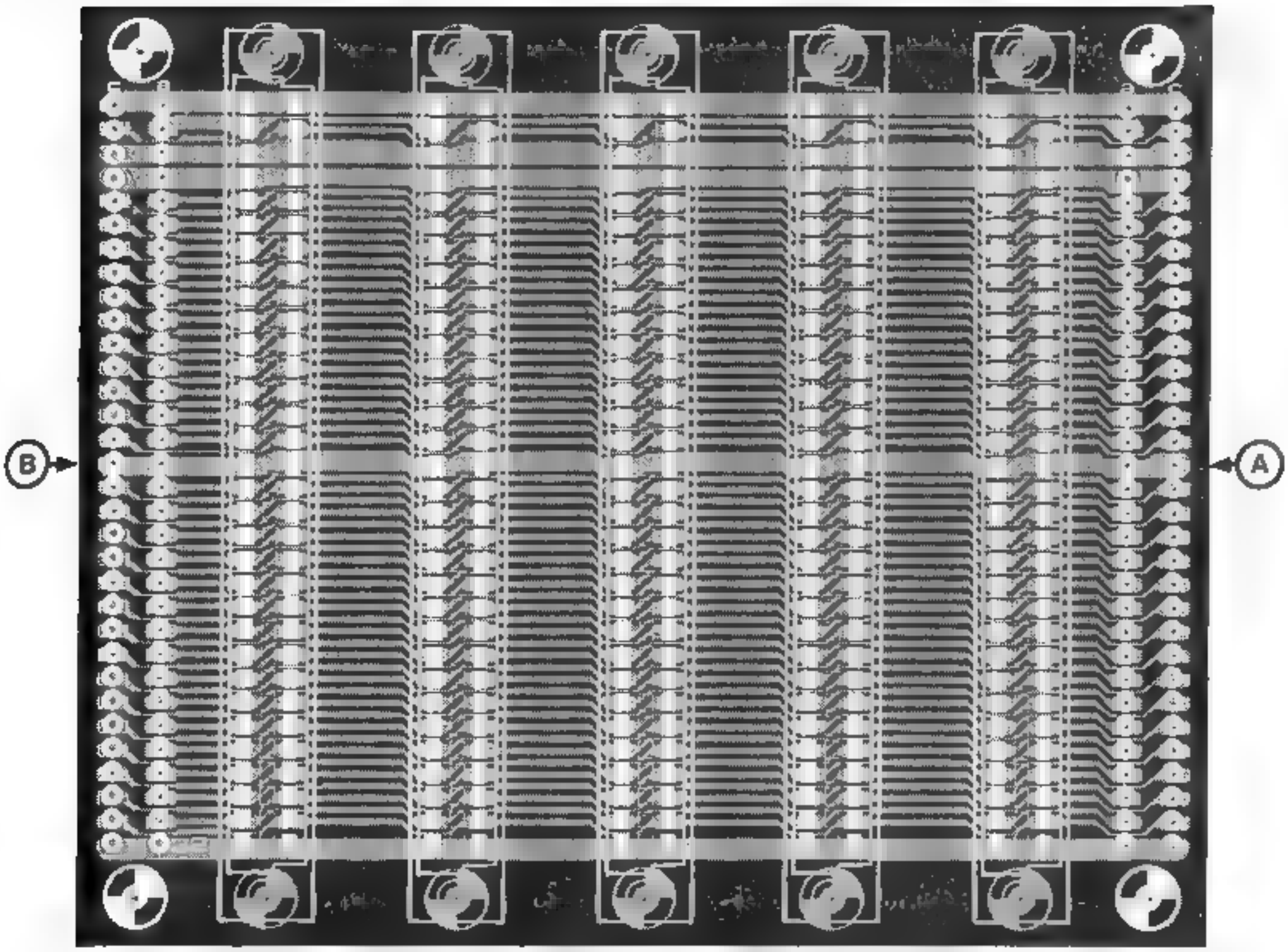
Bild 21a. Ober- und Unterseite der Buskarte; sie ist hier ebenfalls verkleinert wiedergegeben.

Diese "Karten-Architektur" ist selbstverständlich nicht verbindlich, sondern es sind, wie noch deutlich werden wird, zahlreiche Modifikationen möglich. Bei der Auswahl des Gehäuses (siehe Aktionspunkt 5) hat man weitgehend freie Hand. Die beschriebene Anordnung der Karten soll lediglich als Ausgangssituation für die nachfolgenden Betrachtungen dienen, die die verschiedenen Verbindungen von Karte zu Karte betreffen.

Von der Basiskarte zur Interfacekarte

Es gibt mehrere Möglichkeiten für die Verbindung des Erweiterungskonnektors der Basiskarte mit dem Eingangskonnektor der Interfacekarte.

1. Man verwendet hierzu einen Teil der Buskarte. Von ihr (sowie von "Buskartenspeichern") war zwar schon häufiger die Rede, doch bisher wurde sie noch nicht genauer betrachtet. Gemeint ist die Buskarte EPS 80024 des Elektor-SC/MP-Systems, deren Layout zum ersten Mal im Dezember 1979 von der Zeitschrift "Elektor" veröffentlicht wurde. Bild 21 zeigt diese Platine in verkleinertem Maßstab. Von der Busplatine



wird nur der mittlere Teil benötigt, der drei Konnektoren aufnehmen kann; die beiden übrigen Teile werden abgesägt. Der Platz für den mittleren Konnektor bleibt frei, während die beiden übrigen Doppel-Lochreihen die Konnektoren für die Basis- und die Interfacekarte aufnehmen. Bild 20a zeigt diese Steckverbindung in der Seitenansicht. Die beiden Einwände, daß die Busplatine eigentlich zum Zersägen zu schade ist und die Sandwich-Konstruktion dicker als notwendig ausfällt, sind berechtigt. Diese Lösung ist deshalb nur sinnvoll, wenn derartige Gesichtspunkte keine Rolle spielen. Macht man doch von ihr Gebrauch, dann beachte man die Asymmetrie der Kupferbahnen: Die Leiterbahnen der Anschlußpunkte 3 und 4 sind bedeutend breiter als die Leiterbahnen der Punkte 30 und 29, so daß Probleme infolge unzulässiger Spannungsabfälle entstehen können, wenn die Konnektoren falsch auf die Busplatine gesetzt werden.

2. Die Verbindungen werden von einer Lötpunktrasterplatine (Raster 2,54 mm) hergestellt, die man zu einer Buskarte mit zwei Konnektoren ausbaut. Der Abstand zwischen den beiden Konnektoren (siehe ebenfalls Bild 20a) kann den individuellen Erfordernissen angepaßt werden. Ein minimaler Abstand zwischen Basiskarte und Interfacekarte ist möglich, wenn die Schalter S24 und S25 nicht auf der Basiskarte, sondern separat montiert werden. Auf jeden Fall ist aber ein ausreichender Sicherheits-

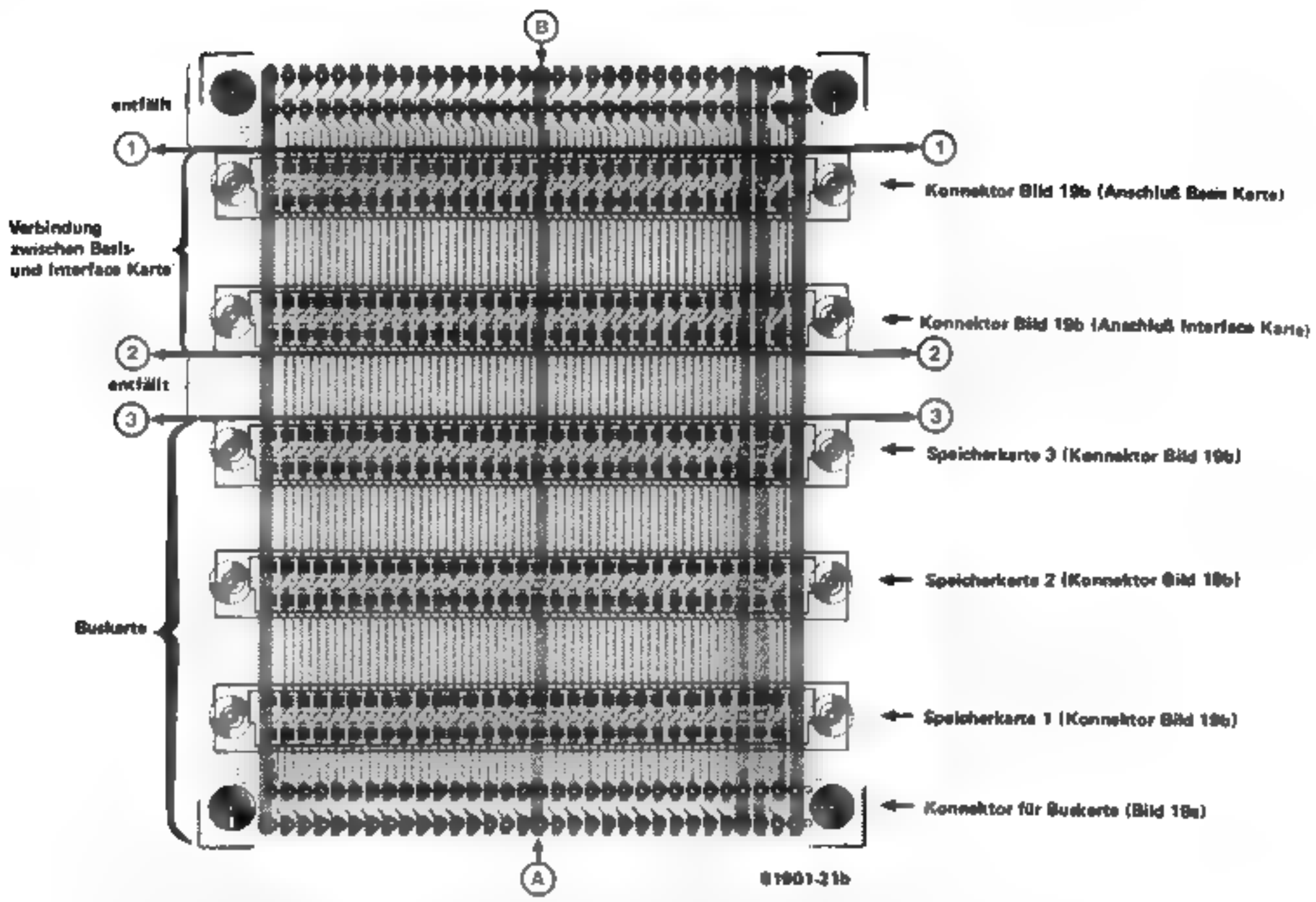


Bild 21b. Wenn nicht mehr als drei Busspeicherkarten geplant sind, kann man eine Buskarte zersägen und einen Teil davon als Verbindung zwischen Basiskarte und Interfacekarte verwenden. Im Bild ist angegeben, wo die Karte durchtrennt wird (①, ② und ③) und welche Konnektoren montiert werden. Falls man eine Laubsäge benutzt, ist mit einem möglichst fein gezahnten Sägeblatt zu arbeiten. Die Konnektoren werden vorher eingelötet, so daß die Leiterbahnen nicht von der Platine abreißen können.

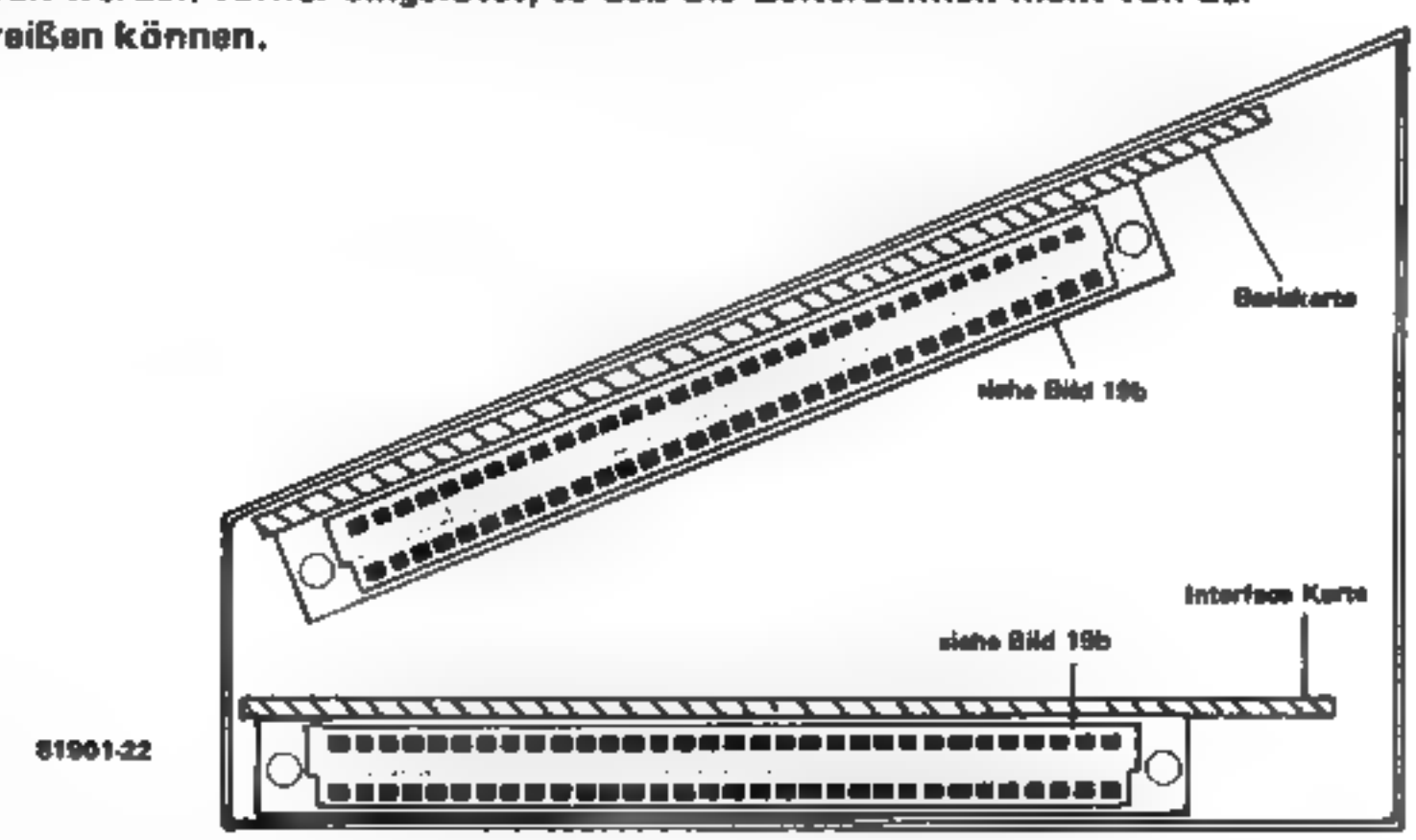


Bild 22. Querschnitt durch ein Pultgehäuse, in das die Basis- und die Interfacekarte nicht als "Sandwich", sondern in spitzem Winkel zueinander eingebaut sind. Die Verbindung der beiden Kartenkonnektoren kann hier eine "Verlängerungsschnur" übernehmen. Das ist ein Stück Flachkabel mit einem Konnektor entsprechend Bild 19b an jedem Ende.

abstand erforderlich, damit zwischen den Platinen keine Kurzschlüsse entstehen.

3. Man verbindet zwei Konnektoren entsprechend Bild 19b über Flachkabel miteinander und steckt diese "Verlängerungsschnur" auf die Konnektoren von Basis- und Interfacekarte auf. Wenn die beiden Karten nicht starr miteinander verschraubt werden, kann man sie einzeln in beliebiger Lage montieren.
4. Man verzichtet völlig auf Konnektoren aller Art und verbindet die Platinen unmittelbar über Flachkabel. Diese Methode spart zwar Kosten, nicht aber unbedingt Zeit und eventuellen Ärger. Da die Anschlußpunkte leicht verwechselt werden können und außerdem ungewöhnlich dicht beieinanderliegen, sollten nur sehr löterfahrene Hobby-Elektroniker zu dieser Lösung greifen.

Übrig bleiben noch die fünf Leitungen zwischen den beiden Sandwichkarten, die das Kassetten- und das RS-232-Interface auf der Interfaceplatine mit dem Portkonnektor auf der Basiskarte verbinden müssen. Entweder lötet man diese Leitungen an die betreffenden Stifte eines 31-poligen Konnektors (männlich; siehe Bild 19d), oder die Drähte werden unmittelbar an die Lötpunkte auf der Basiskarte gelötet. Der PIA-Portkonnektor kann vorher entfernt und später als VIA-Applikationskonnektor weiterverwendet werden. Dieser Konnektor wird auf der Basiskarte nicht mehr benötigt, da der PIA (im Gegensatz zum VIA auf der Interfacekarte) nur noch für interne Aufgaben zuständig ist.

Von der Interfacekarte zur Buskarte

Dieser Abschnitt ist nur für Leser aktuell, die die Speicherkapazität des Junior-Computers über insgesamt 8 K hinaus erweitern wollen. Die Bilder 20b und 20c zeigen die Konnektorverbindungen zwischen Interfacekarte und Buskarte in der Seitenansicht. Als Buskarte kann nicht nur die schon erwähnte Platine EPS 80024 (Bild 21) des Elektor-Platinenservice dienen, sondern auch hier ist eine mit entsprechenden Verbindungen versehene Lötpunktrasterplatine (sogenannte Experimentierplatine) verwendbar. In beiden Fällen werden daneben noch ein männlicher Konnektor mit abgewinkelten Anschlüssen (Bild 19a) und ein bis fünf weibliche Konnektoren mit geraden Anschlüssen (Bild 19b) benötigt. Bei Verwendung der EPS-Busplatine ist auf die Asymmetrie der Leiterbahnen zu achten! Die Lötpunkte 1 auf der Busplatine, deren Lage aus dem Aufdruck hervorgeht, müssen auf jeden Fall mit den Lötpunkten 1 auf der Interfaceplatine verbunden sein.
Für alle 64-poligen Konnektoren der zum Sandwich verschraubten Basis- und Interfacekarte gilt, daß die Lötpunktreihe a näher am Platinenrand liegt als die Lötpunktreihe c. Bei der Buskarte ist dies umgekehrt: die c-Reihe hat den geringeren Abstand zum Platinenrand. Deshalb finden sich die Signale, die auf der Interfacekarte an den a-Anschlüssen des Eingangskonnektors liegen, an den entsprechenden c-Anschlüssen des Ausgangskonnektors wieder, und umgekehrt (vergleiche Schaltung in Bild 1).
Anmerkung: Alle Bezeichnungen a und c der Kontaktreihen in Bild 19 sind identisch mit den Angaben auf den Konnektoren; im Fall der Buskarte stimmen diese nicht mit dem Platinenaufdruck überein!

Verbindet man die Interfacekarte und die Buskarte über 64 Einzeldrähte, sind die Drähte a ebenso lang wie die Drähte c (Bild 20d). Für die Montage des Ausgangskonnektors gibt es zwei Möglichkeiten (Bild 20b und 20c), die mit der schon besprochenen Wahl des Konnektortyps zusammenhängen: auf die Platine oder gegen den Platinenrand. Im ersten Fall (Bild 19a) ergibt sich zwischen Interface- und Buskarte ein Höhenunterschied, im zweiten Fall (Bild 19c) liegen beide Karten in der gleichen Ebene.
Wenn die Oberseite des Gehäuses, wie in Bild 22 skizziert, schräg nach vorn geneigt ist und die Interfacekarte parallel zur Basiskarte montiert wird, ist die in Bild 19e dargestellte Verbindungsart eine Lösung. Die Buskarte soll nämlich entweder waagerecht oder senkrecht stehen. Um hier den Winkel zwischen Interface- und Busplatine auszugleichen, können aber auch Konnektoren zusammen mit einem flexibelen "Verbindungskabel" verwendet werden.
Eine senkrechte Stellung der Buskarte ist dann von Vorteil, wenn man einem flacheren, jedoch dafür tieferen Gehäuse den Vorzug gibt. In diesem Fall ist, verglichen mit der waagerechten Position, die gegenüberliegende Seite der Busplatine die Schnittstelle zur Interfacekarte. Bild 20f zeigt die Verbindung in der Seitenansicht. Auch hier ist unbedingt die Asymmetrie der Buskarte zu beachten; die Punkte 1 (Platinenaufdruck) müssen bei der Interfacekarte und der Buskarte übereinstimmen. Es ist nicht zulässig, daß beispielsweise die Punkte 1 der Interfacekarte mit den Punkten 32 der Busplatine verbunden sind.
Die Zahl der möglichen Varianten für die Konnektor- oder Drahtverbindungen ist damit noch längst nicht erschöpft. Aus den behandelten Beispielen wurde aber sicher deutlich, worauf es bei der "Verdrahtung" ankommt.

Aktionspunkt 5: Das Gehäuse

separat oder kompakt?

Während in ländlichen Gegenden die zu einem Hof gehörenden Gebäude oft verstreut auf einem Grundstück stehen, befindet sich in Stadtregionen meistens alles unter einem Dach. Ähnlich kann es sich beim erweiterten Junior-Computer oder sogar beim Standard-Junior-Computer verhalten. Beispielsweise beansprucht die Speisung relativ viel Platz. Warum soll sie nicht ihr eigenes Gehäuse erhalten? Das Gehäuse des Computers selbst ist dann kompakt und handlich. Dies ist eine Möglichkeit. Die andere, die ebenfalls ihre Vorteile hat, ist folgende: Der Computer wird, abgesehen von den peripheren Geräten (Kassettenrekorder, Drucker usw.), in einem einzigen Gehäuse untergebracht. Zwar muß dieses Gehäuse beträchtlich mehr Raum bieten, doch das Gewirr von Verbindungskabeln, das im ersten Fall notwendig war, wird drastisch entflechtet.
Im Zuge der Erweiterungen steigt auch die Anzahl der peripheren Geräte: ein oder sogar zwei Kassettenrekorder, die alfanumerische Tastatur, das Videoterminal mit dem TV-Gerät. Natürlich läßt sich ein großes Gehäuse realisieren, das sämtliche Systemkomponenten beherbergen kann. Da ein solcher "Schrank" aber doch reichlich unhandlich ist, wird man in der

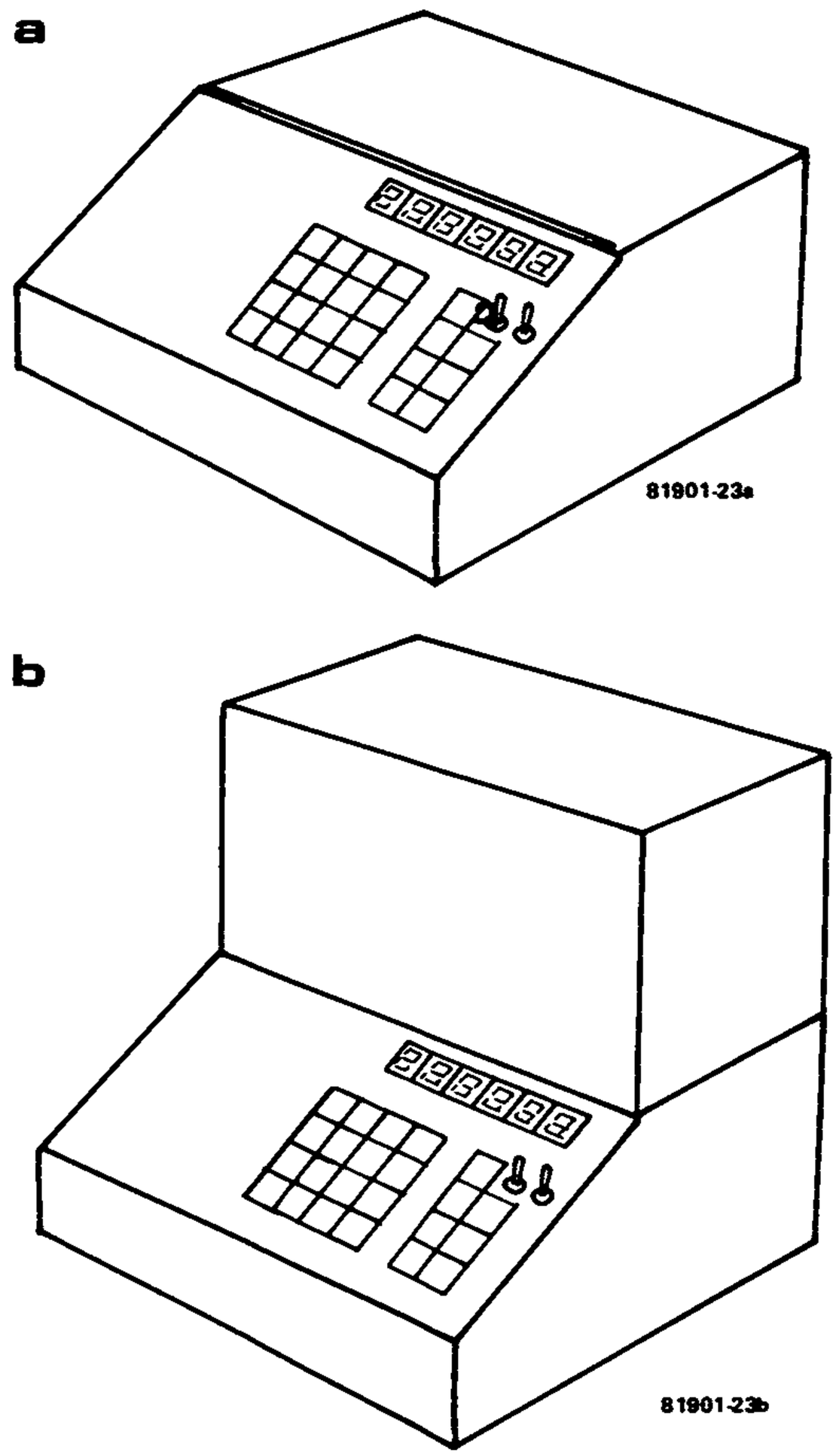


Bild 23. Zwei Gehäusetypen, die für den Einbau des Junior-Computers oder eines Teils davon geeignet sind. Das sogenannte Verstärkergehäuse (b) hat gegenüber der Pultform (a) den Vorteil, daß es mehr Platz bietet. Ob "alles unter einem Dach" oder zum Beispiel das Netzteil separat untergebracht wird, hängt von individuellen Gesichtspunkten ab.

Regel bestimmte Systemkomponenten separat aufstellen. Das TV-Gerät und die Kassettenrekorder lassen sich dann leichter nebenbei noch für andere Zwecke nutzen. Falls der Standard-Junior-Computer bereits in einem eigenen Gehäuse untergebracht ist, bietet sich auch die Anschaffung eines zweiten Gehäuses für das erweiterte Netzteil und die Busspeicherkarten an.
Zwischen den beiden Extremen der "Viel-Gehäuse-Konzeption" und des "Super-Kompakt-Aufbaus" liegen zahlreiche Varianten, von denen folgende Lösung wohl am naheliegendsten ist: In einem gemeinsamen Gehäuse werden das Netzteil und sämtliche Karten (einschließlich eventueller Bus-

speicherkarten) untergebracht, während die peripheren Komponenten ein "Eigenleben" führen. Unter den Gehäuseformen sind insbesondere das Pultgehäuse (Bild 23a) und das sogenannte Verstärkergehäuse (Bild 23b) geeignet. Letzteres erhielt seinen Namen während des Röhrenzeitalters, als die Endröhren von Niederfrequenz-Leistungsverstärkern einen beträchtlichen Raum über dem flachen Chassis einnahmen. Die Technologie der Endverstärker hat sich längst gewandelt, die Gehäuseform blieb jedoch erhalten. Dieser Gehäusetyp eignet sich auch für unseren Zweck. Unter dem schrägen Bedienpult findet das Karten-Sandwich seinen Platz, während der hintere hohe Teil die Busspeicherkarten (links) und die erweiterte Speisung mit den beiden Netztrafos (rechts) aufnimmt.
Ein flaches Pultgehäuse (Bild 23a) wird meistens dann angeschafft worden sein, wenn bereits der Standard-Junior-Computer eine Behausung erhielt. Dieser Gehäusetyp hat eine weniger wuchtige Gestalt als das Verstärkergehäuse. Er ist dann von Vorteil, wenn man das Karten-Sandwich, das erweiterte Netzteil und die Busspeicherkarten in separaten Gehäusen unterbringen will. Hier braucht man sich bei der Auswahl des Gehäuses noch nicht für oder gegen die einzelnen Erweiterungen entscheiden; sie können später nach dem Baukastenprinzip hinzugefügt werden.
Zur Auswahl des Gehäuses und zum Einbau der diversen Komponenten folgende Hinweise:

1. Die Sandwich-Montage von Basis- und Interfacekarte ist anderen Montagearten, beispielsweise der Anordnung der Karten nebeneinander, unbedingt vorzuziehen.
2. Es ist technisch möglich, die Verbindungen zwischen den einzelnen Karten (Basiskarte/Interfacekarte; Interfacekarte/Buskarte) über Flachkabel von maximal 50 cm Länge herzustellen.
3. Fällt die Wahl auf das flache Pultgehäuse, eventuell, weil es schon vorhanden ist, und sollen Busspeicherkarten angeschlossen werden, so kann der Ausgangskonnektor auf der Gehäuserückseite montiert werden. Die Busspeicherkarten können dann unmittelbar oder, falls sie in ein eigenes Gehäuse eingebaut sind, über eine "Verlängerungsschnur" angeschlossen werden. Für die Verbindung werden zwei Konnektoren und ein Stück Flachkabel (Länge maximal 50 cm) benötigt.
4. Man sorge dafür, daß alle Karten mechanisch stabil und erschütterungssicher montiert sind. Die Karten dürfen nicht ausschließlich durch die Konnektoren gehalten werden, sondern sind in geeigneter Weise auch an anderen Punkten zu stützen. Für senkrecht auf einer anderen Fläche stehende Karten sind nach Möglichkeit Führungsschienen zu montieren. Bei Verwendung von Abstandsröllchen aus Metall achte man auf mögliche Kurzschlüsse zwischen den angrenzenden Platinenbahnen oder von der Platine zum Gehäuse. Falls zwei Karten wie in Bild 20b skizziert nicht in der gleichen Ebene liegen, muß der Höhenunterschied durch Unterlegscheiben oder ähnliches ausgeglichen werden.
5. Zum Schluß noch eine Art "Checkliste", in der alles das aufgeführt ist, was bei der Auswahl und der Bearbeitung des Gehäuses eine Rolle spielt:

a. Das hexadezimale Tastenfeld.
b. Das sechsteilige Siebensegmentdisplay. Es wird wie bisher benötigt, auch wenn man überwiegend mit der ASCII-Tastatur, dem Video-

terminal und dem Drucker arbeitet.

c. Der Step-Schalter S24.

d. Der Displayschalter S25. Diesen Schalter und den Stepschalter kann man eventuell von der Basisplatine demontieren und an anderer Stelle unterbringen. Das Kartensandwich ist dann so flach, daß es in praktisch jedes Pultgehäuse paßt.

e. Der vorhandene 31-polige PIA-Portkonnektor. Da jetzt der VIA auf der Interfacekarte die Verbindungen zur Außenwelt herstellt, kann man diesen Konnektor demontieren.

f. Der Erweiterungskonnektor. Vielleicht wurde dieser Konnektor bereits auf dem Gehäuse des Junior-Computers in seiner Basisversion montiert, mit der Absicht, die Interfacekarte extern anzuschließen. Dies ist natürlich möglich, die empfohlene Sandwich-Konzeption würde dabei jedoch verlassen.

g. Die Einführung des Netzkabels (oder eine Kaltgerätebuchse), der Netzschalter und der Sicherungshalter. Hier muß beim erweiterten Junior-Computer eine 2-A-Sicherung eingesetzt werden!

h. Der RS-232-Konnektor. Er wird direkt auf die Interfacekarte oder auf die Gehäuserückwand (Verbindung zur Interfacekarte über Flachkabel) montiert. Im zweiten Fall ist bei der Verdrahtung auf die richtige Reihenfolge der Pins zu achten. Die Pin-Numerierung 1 . . . 25 bezieht sich auf die Montage des Konnektors auf die "Kupferseite" der Interfacekarte. Beim Anschluß des Elekterminals können die Brücken zwischen bestimmten Pins (siehe Schaltung in Bild 2) entfallen.

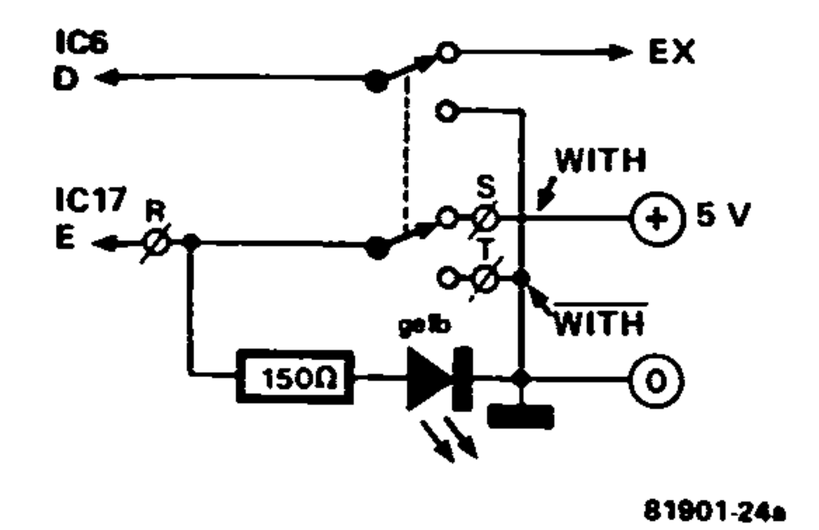
i. Die Buchsen J1 . . . J4 für die Leitungen zu den Kassettenrekordern. Am zweckmäßigsten dürfte ihre Montage auf der Gehäuserückseite sein. Verwendbar sind natürlich auch DIN-Buchsen; J3 und J4 werden nur bei Vorhandensein von Fernbedienungsanschlüssen an den Rekordern benötigt.

j. Die Kassetteninterface-LEDs erhalten einen Platz auf dem Bedienpult. Entweder verwendet man Gehäuse-LEDs mit Zentralbefestigung, oder gewöhnliche LEDs werden mit Hilfe von Clips (Fassungen) montiert.

k. Die beiden Netzteil-LEDs. Bei Unterbringung des Netzteils in einem separaten Gehäuse bringt man sie dort an, sonst setzt man sie auf das Bedienpult in die Nähe des Netzschalters.

l. Nur bei separatem Netzteil: fünf Buchsen für die Speisespannungen einschließlich Masse. Man spare nicht am Kupferquerschnitt der Verbindungsleitungen; dies gilt insbesondere für die Masse- und die +5 V-Leitung. Flexibele 1 mm^2-Leitungen mit verschiedenfarbiger Isolierung schützen vor unzulässigem Spannungsabfall und vor Verwechslung der Anschlüsse. Um störenden Beeinflussungen von außen vorzubeugen, können die spannungsführenden Buchsen mit Tantalelkos 1 μ/16 V zur Massebuchse hin abgeblockt werden. Polarität der Tantalelkos beachten!
Die fünf zentralen Anschlußpunkte für die Speisespannungen und für Masse befinden sich auf der Interfacekarte. Die Basiskarte und die Bus-speicherkarten werden über die Konnektoren versorgt.

m. Der VIA-Konnektor. Er übernimmt die Rolle des PIA-Portkonnektors der Basisversion und wird abhängig von den geplanten Applikationen des erweiterten Junior-Computers entweder auf die Gehäuserückwand oder auf das Bedienfeld gesetzt.



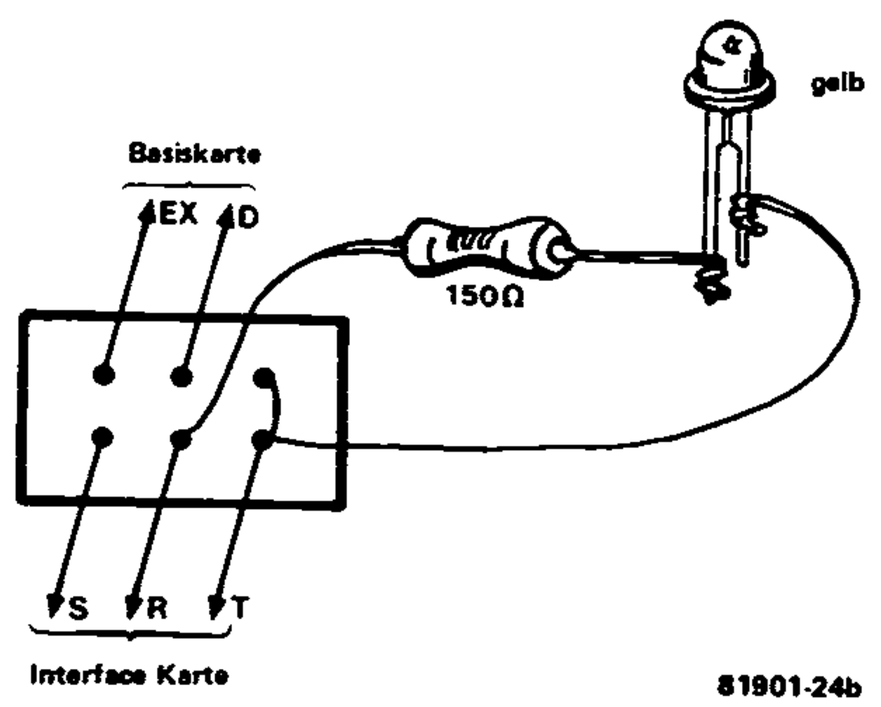


Bild 24. Wenn die Busspeicherkarten nicht fest installiert sind, sondern beispielsweise von außen eingesteckt werden können, muß auf dem Bedienpult ein Umschalter vorhanden sein. Das Bild zeigt, wie dieser Schalter anzuschließen ist.

n. Der Ausgangskonnektor. Nur wenn Buskartenspeicher extern angeschlossen werden sollen, ist dieser Konnektor notwendig. Hier wird entweder die Buskarte unmittelbar eingesteckt, oder die externe Speichereinheit wird über eine "Verlängerungsschnur" (Flachkabel und zwei männliche Konnektoren) angeschlossen.

o. Der Schalter WITH/WITHOUT; er dient zum Umschalten der Verbindungen R-S & D-EX nach R-T & D-⊥ (Interface- bzw. Basisplatine). Dieser Schalter ist nur erforderlich, wenn man den Junior-Computer wahlweise mit oder ohne externe Busspeicherkarten betreiben will. Bild 24 zeigt den Anschluß des Schalters. In Stellung "WITH" leuchtet eine gelbe LED auf, die neben dem Schalter montiert wird.

Damit sind wir am Ende der Baubeschreibung für den erweiterten Junior-Computer angelangt. Natürlich konnten nicht alle überhaupt denkbaren Modifikationen und Sonderfälle berücksichtigt werden; der Spielraum, den der mechanische Aufbau des Juniors zuläßt, ist zu groß. Doch dies ist, so meinen wir, eher ein Vorteil als ein Nachteil. Nach Abschluß der "Bauarbeiten" wollen wir uns nun den notwendigen Endkontrollen und Tests sowie schließlich der Inbetriebnahme zuwenden.

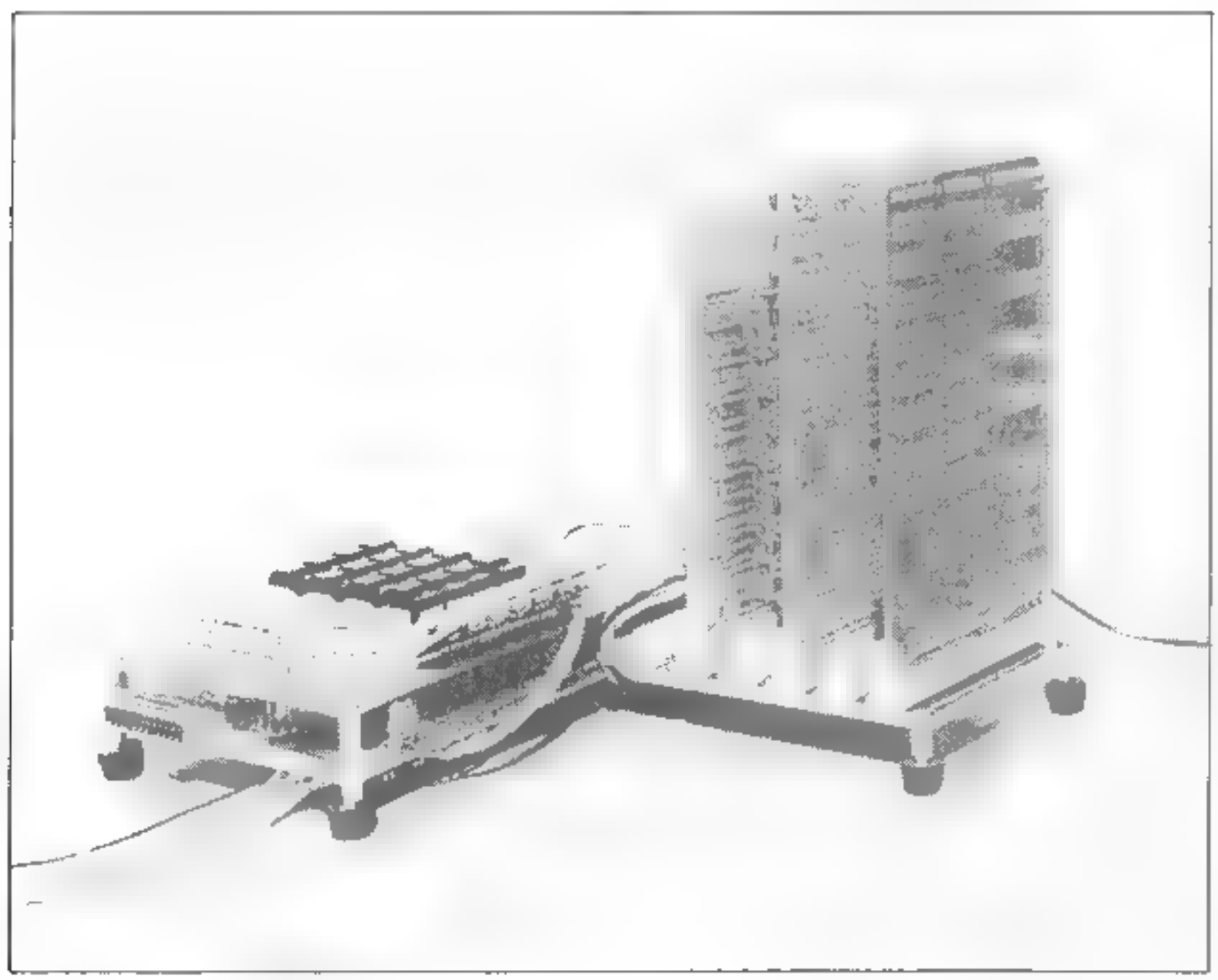


Foto 1. Der erweiterte Junior-Computer, betrachtet aus "nordöstlicher" Richtung. (Zur Orientierung: der Benutzer, so stelle man sich vor, habe bei der Bedienung das Gesicht nach Norden gerichtet.) Der RS-232-Konnektor ist ebenso wie die INPUT- und OUTPUT-LED unmittelbar auf die Interfacekarte montiert.

Aktionspunkt 6: Kontrollen, Tests, Inbetriebnahme und – falls wider Erwarten doch notwendig – Fehlersuche

Oder: Wie man dem erweiterten Junior-Computer das Laufen beibringt.

Die Endkontrolle ist sozusagen der krönende Abschluß der Arbeiten mit Lötkolben, Pinzette und Seitenschneider. Wir hoffen, daß sie nur wenig Zeit in Anspruch nimmt und eine Fehlersuche dank der bisherigen Sorgfalt und Umsicht entfällt. Die Überprüfungen lassen sich in zwei Gruppen einteilen: in hardware- und softwareorientierte Kontrollen. Es sei noch einmal an den Vergleich Mensch-Computer erinnert, bei dem Parallelen zwischen der Hardware und dem Fleisch und Blut sowie zwischen der Software und dem Geist des Menschen gezogen wurden. Während beim Menschen der Körper und der Geist voneinander unabhängig gesund und krank sein können, lassen sich bei dem weniger komplizierten Computer durchaus Rückschlüsse vom einen zum anderen Bereich ziehen. Wenn beispielsweise ein Softwaretest ergibt, daß in einem neuen RAM-Bereich geschrieben und gelesen werden kann, dann muß auch die zugehörige Hardware in Ordnung sein. Durch geeignete Softwaretests kann man auch

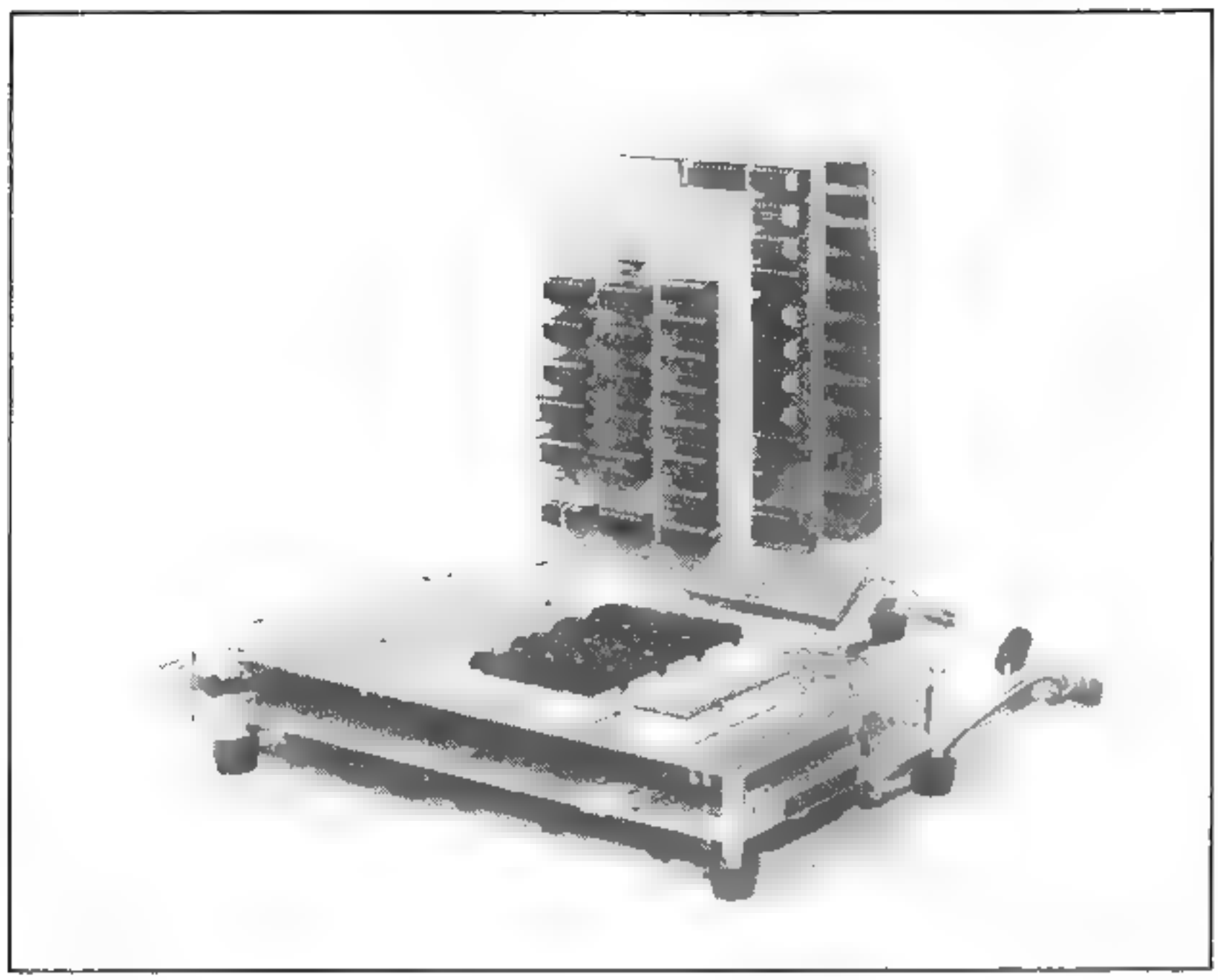


Foto 2. Wie Foto 1, jedoch Ansicht aus "südöstlicher" Richtung. Man beachte, daß die Kupferseite der Interfacekarte nach oben zeigt. Für den Anschluß der Kassettenrekorder wurden Cinch-Buchsen verwendet. Die erste Speicherkarte ist ein Laboraufbau der 16-K-RAM-Karte, bestückt mit dynamischen RAMs.

einen eventuellen Hardwarefehler einkreisen, um ihn schließlich zu finden und zu beseitigen.

Nachdem nun der erweiterte Junior-Computer aufgebaut ist, stehen zunächst hauptsächlich hardwareorientierte Kontrollen auf dem Programm. Sie sind notwendig, lange bevor die Speisespannungen angeschlossen werden können und der erweiterte Junior-Computer eingeschaltet werden darf.

Die in Buch 1 auf den Seiten 35 und 36 beschriebenen allgemeinen Überprüfungen sind auch nach der Erweiterung noch einmal vorzunehmen. Wir gehen davon aus, daß die Standard-Version des Juniors lief und die beim Bau und Test gewonnenen Erfahrungen auch hier einfließen. Deshalb wollen wir uns nachfolgend auf die wichtigsten allgemeinen Ratschläge und die Besonderheiten der Erweiterung beschränken.

Leitungsverbindungen. Es ist tatsächlich in der Praxis so, daß die Gefahr von falschen oder fehlenden Verbindungen mit zunehmender Anzahl der Leitungen sprunghaft ansteigt. Da im erweiterten Junior-Computer nicht nur ein Vielfaches an Bauelementen, sondern auch an Lötstellen, Konnektoranschlüssen, Drahtverbindungen usw. steckt, steht die Suche nach fehlenden oder falschen Verbindungen (Kurzschlüsse, kalte Lötstellen,

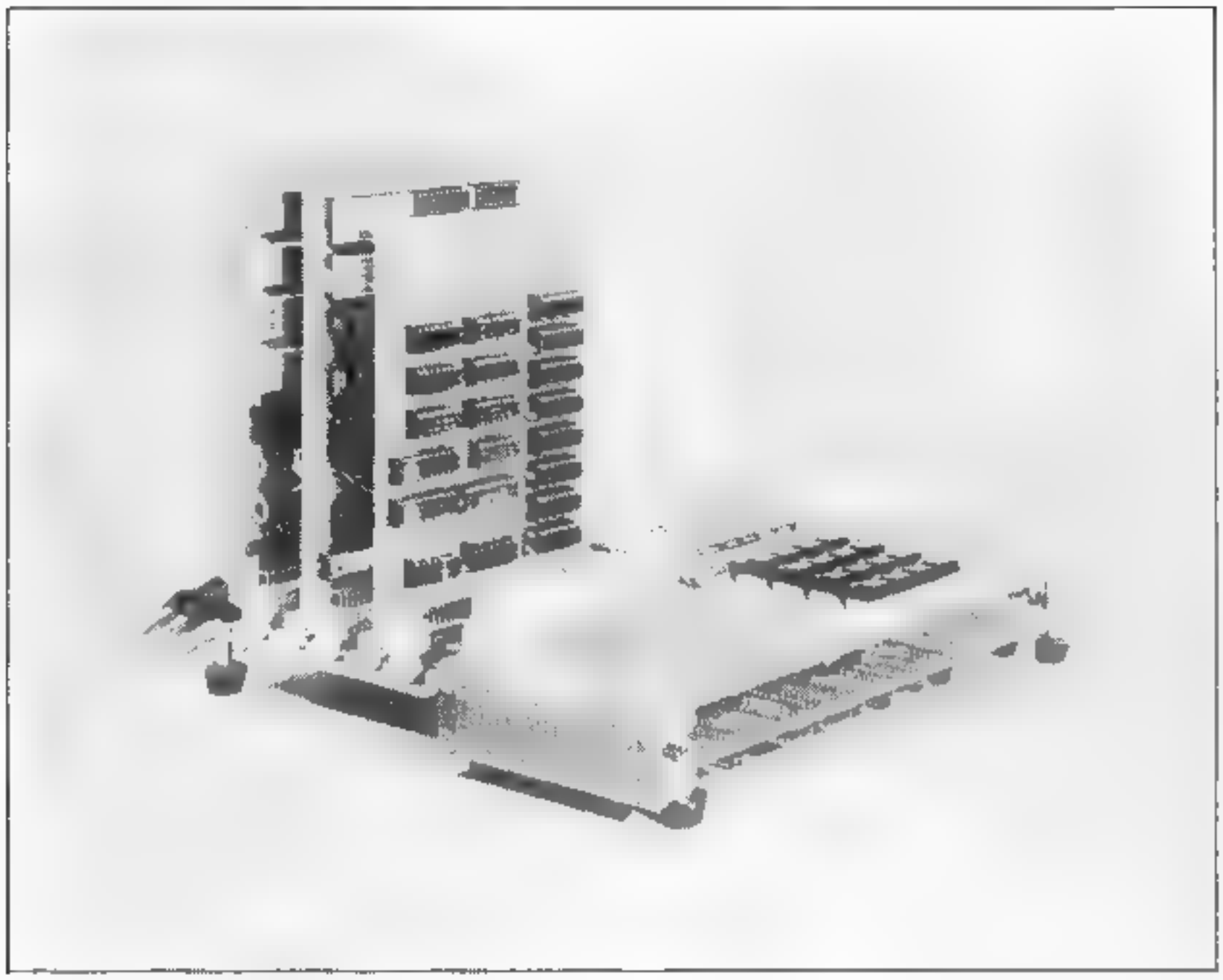


Foto 3. Wie Foto 1, jedoch Ansicht aus "südwestlicher" Richtung. Die Verbindung zwischen dem Erweiterungskonnektor der Basiskarte und dem Eingangskonnektor der Interfacekarte stellt ein Stück der Busplatine (siehe Bild 21b) her. Außerdem sind die Verbindungen mit dem Portkonnektor auf der Basiskarte über einen eigentlich für Platinenmontage bestimmten 31-poligen Konnektor sichtbar.

Verwechslung von Leitungen und Anschlußpunkten) an erster Stelle auf der Checkliste. Deshalb müssen möglichst viele Verbindungen mit einem Ohmmeter oder mit dem im Juni-Heft '81 der Zeitschrift Elektor beschriebenen "Durchgangspiepser" überprüft werden. Haben alle IC-Pins einwandfreien Kontakt? Wurden keine Anschlußdrähte verwechselt oder an falsche Punkte gelötet? Gibt es bei den diversen männlichen und weiblichen Konnektoren keine "Kontaktschwierigkeiten"? Nehmen sämtliche Drahtbrücken die richtige Lage ein?

Bauelemente. Man betrachte mit dem geschärften Blick eines Detektivs nacheinander die Bauelemente, die zwar in mehreren Positionen angebracht werden können, aber nur eine bestimmte Position einnehmen dürfen. Dazu gehören ICs, Elkos, Dioden, Transistoren und der 25-polige D-Konnektor (RS232). Auch die Werte von Widerständen werden hin und wieder verwechselt (Farbcode!). Lieber einmal zu viel kontrolliert als einmal zu wenig.

Wurde wirklich alles überprüft, was sich vor Einschalten der Speisespannungen überprüfen läßt? Vielleicht kann auch ein anderer Junior-Computer-Enthusiast aus dem Bekanntenkreis oder aus einem Anwender-

Foto 4. Dies meinen wir, wenn wir schreiben: "Nicht nur die Position des ICs muß stimmen, sondern auch die Lage aller IC-Pins." Ein weiterer Kommentar ist sicherlich überflüssig.

Club die Endkontrolle übernehmen. Wenn alles in Ordnung befunden wurde, ist es an der Zeit, die ebenfalls getestete Speisung über fünf verschiedenfarbige, flexible, einadrige Leitungen anzuschließen. Die Masse- und die +5 V-Leitung haben einen Querschnitt von mindestens 1 mm^2. Sind diese Leitungen dünner, so können ungeahnte Überraschungen die Folge sein; der Junior-Computer arbeitet dann höchstwahrscheinlich aus "unerklärlichen" Gründen fehlerhaft. Obwohl es eigentlich selbstverständlich ist, sei es noch einmal betont: **Die richtigen Spannungen müssen an den richtigen Punkten auf der Interfacekarte angeschlossen sein!**
Wenn alles in Ordnung befunden wurde, schreiten wir zur Tat. Jetzt ist nämlich der spannende Moment gekommen: der erweiterte Junior-Computer wird eingeschaltet. Anschließend drücken wir als erstes die RST-Taste. Das Aufleuchten der Siebensegmentdisplays signalisiert, daß zumindest der Basisteil des erweiterten Junior-Computers arbeitet. Danach kann man die verschiedenen Speisespannungen an den IC-Pins (siehe Bild 18) messen. **Die Spannungen müssen unmittelbar am IC-Gehäuse und nicht an den Lötpunkten der Platinenkupferseite gemessen werden.** Weshalb? Weil nur so eine Unterbrechung oder ein zu hoher Übergangswiderstand zwischen Platinenleiterbahn und IC-Pin festgestellt werden kann. Zwei benachbarte IC-Pins dürfen beim Messen nicht kurzgeschlossen werden; dies kann eine Beschädigung des ICs zur Folge haben. Leuchtet

nach Drücken der RST-Taste das Display nicht auf, während die Basisversion des Juniors vorher einwandfrei arbeitete, dann sind folgende Fragen zu klären: Erhält die Basiskarte die von ihr benötigten Speisespannungen? Wie ist die Drahtbrücke an Eingang D von IC6 angeschlossen? Sie muß ohne Busspeicherkarten an Masse und mit dieser Speichererweiterung an EX liegen. Im zweiten Fall: Ist Seite FF mit einem EPROM belegt und enthält dieses EPROM unter den Adressen FFFC und FFFD den Resetvektor (FFFC: 1D; FFFD: 1C)? Wurde die Drahtbrücke R-S eingelötet? Arbeitet der Junior-Computer, wenn vorübergehend die Drahtbrücken zwischen D und Masse sowie zwischen R und T angebracht werden? Oder arbeitet er, wenn die Basiskarte herausgezogen wird und über die früheren Anschlüssen ihre Speisespannungen erhält? Zu überprüfen sind insbesondere die Interface-Hardware in der Umgebung der Adressendekodierung und der Datenbussteuerung einschließlich des PROM IC17. Ist dieses PROM tatsächlich richtig programmiert?

Wenn der im EPROM auf der Basiskarte untergebrachte Monitor arbeitet, ist mit ihm ein wirkungsvolles Mittel zur Durchführung weiterer Tests zur Hand. Zum Beispiel läßt sich leicht die richtige Funktion der hinzugekommenen RAMs testen: IC2 und IC3 sowie IC4 und IC5, falls für die letzten beiden ICs der Typ 8114 gewählt wurde. Man gibt über AD eine Adresse ein, die zum Bereich des zu testenden RAMs gehört. Diese Adresse erscheint zusammen mit willkürlichen Daten auf dem Siebensegmentdisplay. Mit DA werden anschließend andere Daten eingegeben. Zeigt die Anzeige die neuen Daten an, dann werden in den gewählten Speicherplatz einwandfrei Daten geschrieben und aus diesem auch wieder ausgelesen (siehe Buch 2, Kapitel 7). In gleicher Weise können beliebig viele Speicherplätze des gleichen oder eines anderen RAM-Adressenblocks getestet werden.

EPROMs werden vom Computer bekanntlich nur gelesen; geschrieben wird ihr Inhalt während der Programmierung mit dem Programmiergerät. Befinden sich die Systemprogramme TM in IC4 und PM in IC5, dann läßt sich mit Hilfe des Monitors und des Hexdump von TM und PM am Schluß dieses Buchs überprüfen, ob der Inhalt korrekt gelesen wird.

Auf einwandfreie Funktion zu überprüfen sind ferner der VIA, das Kassetten- und das RS-232-Interface. Bevor die Kassettenhardware betriebsfähig ist, muß erst noch mit Trimmpoti P1 die Grundfrequenz des PLL eingestellt werden. Darüber jedoch mehr in Kapitel 11. Der VIA ist erst in Buch 4 an der Reihe, und über die richtige oder falsche Arbeitsweise von RS-232- Peripherie läßt sich erst entscheiden, wenn ein solches Gerät gebaut und angeschlossen ist. Das aber wird erst in Kapitel 12 behandelt.

Der erweiterte Junior-Computer ist nun aufgebaut und getestet. Einige Punkte, die für die Inbetriebnahme wichtig sind, kommen noch in Kapitel 11 und 12 zur Sprache. Wenn die Speicherkapazität durch Busspeicherkarten über die 8-K-Grenze ausgedehnt wurde, ist noch zu klären, wie die RAM/EPROM-Karten die ihnen zugedachte Aufgabe erfüllen können. Wie steht es zum Beispiel mit der Adressenorganisation innerhalb des oberen 56-K-Bereichs? Die mit der RAM/EPROM-Speicherkarte zusammenhängenden Fragen beantwortet der nun folgende letzte Teil dieses Kapitels.

Die RAM/EPROM-Speicherkarte (Busspeicherkarte)

Datensammelstelle mit Busanschluß

Die RAM/EPROM-Karte ist ebenso wie die dynamische 16-K-RAM-Karte (die nicht in diesem Buch beschrieben wird) eine Komponente, mit der die Speicherkapazität des erweiterten Junior-Computers um maximal 56 K (Seiten 2Ø . . . FF) erhöht werden kann. Dazu werden eine bzw. mehrere RAM/EPROM-Karten auf die Buskarte gesteckt, die ihrerseits mit der Interfacekarte verbunden ist. Ferner läßt sich bei der Standardversion des Juniors auch eine einzelne RAM/EPROM-Karte unmittelbar an die Buskarte anschließen. Über diese Möglichkeit der Erweiterung gibt der Anhang 1 am Schluß des Buchs Auskunft.
Die Schaltung und der Bau der RAM/EPROM-Karte wurden bereits in einem Artikel der Zeitschrift Elektor besprochen, der im September 1980 erschien. Der Artikel soll an dieser Stelle nicht noch einmal vollständig abgedruckt werden, sondern wir legen hier den Schwerpunkt auf die in Zusammenhang mit dem Junior-Computer wichtigen Aspekte.
Die Karte bietet Platz für maximal vier EPROMs; entweder nimmt sie 1 . . . 4 EPROMs des Typs 2708, 1 . . . 4 x 2716 oder 1 . . . 4 x 2732 auf. Es ist jedoch nicht möglich, unterschiedliche EPROM-Typen auf die gleiche Karte zu setzen. Da der Anschaffungspreis für das 2732 (Kapazität 4 K) noch ungewöhnlich hoch liegt, bleibt dieser EPROM-Typ nachfolgend außer Betracht. Das 2716 bietet dagegen ein bedeutend günstigeres Preis-Leistungs-Verhältnis: es besitzt den doppelten Inhalt eines 2708, sein Preis beträgt jedoch weniger als das Doppelte. Ferner unterscheidet sich das 2716 vom 2708 durch eine einfachere Programmierung und das Fehlen der Speisespannungen –5 V und +12 V.
Die Schaltung der RAM/EPROM-Karte ist in Bild 25 zu finden, während Bild 26 die verkleinerte Platine (EPS 80120) zeigt. Wiedergegeben ist hier nur die Bestückungsseite. Das Original-Layout dieser 10 cm x 16 cm großen Platine wurde in der Zeitschrift Elektor vom September 1980, Seiten 9-28 und 9-29, abgedruckt. Bild 27a zeigt als Ergänzung die Pinbelegung und die interne Schaltung der Dekoder-ICs (IC5 . . . IC7), Bild 27b enthält die zugehörigen Wahrheitstabellen.

Die RAM-EPROM-Karte als Busspeicherkarte für den Junior

Solange kein Speicherbaustein auf der RAM/EPROM-Karte adressiert ist oder in einen adressierten Speicherbaustein Daten geschrieben werden, sind die Datenbuspuffer der betreffenden Karte in Schreibrichtung aktiv. Die in Leserichtung zeigenden Datenbuspuffer werden nur dann aktiviert, wenn aus einem auf der Karte befindlichen RAM oder EPROM Daten gelesen werden. Eine Verfälschung von Daten durch überlagerte Signale (siehe Bild 6) kann daher bei richtiger Dekodierung der Speicherbausteine nicht auftreten.
In Tabelle 9 sind die möglichen Ein- und Ausgangssignale von Hauptadressendekoder IC5 zusammengefaßt (siehe auch Bild 27b). Dieser Dekoder hat die Aufgabe, einen von sechzehn 4-K-Blöcken zu selektieren. Auf der Platine ist zwischen der IC-Pin-Numerierung von IC5 und den Platinenanschlußpunkten (Ø . . . F) zu unterscheiden; beides darf nicht

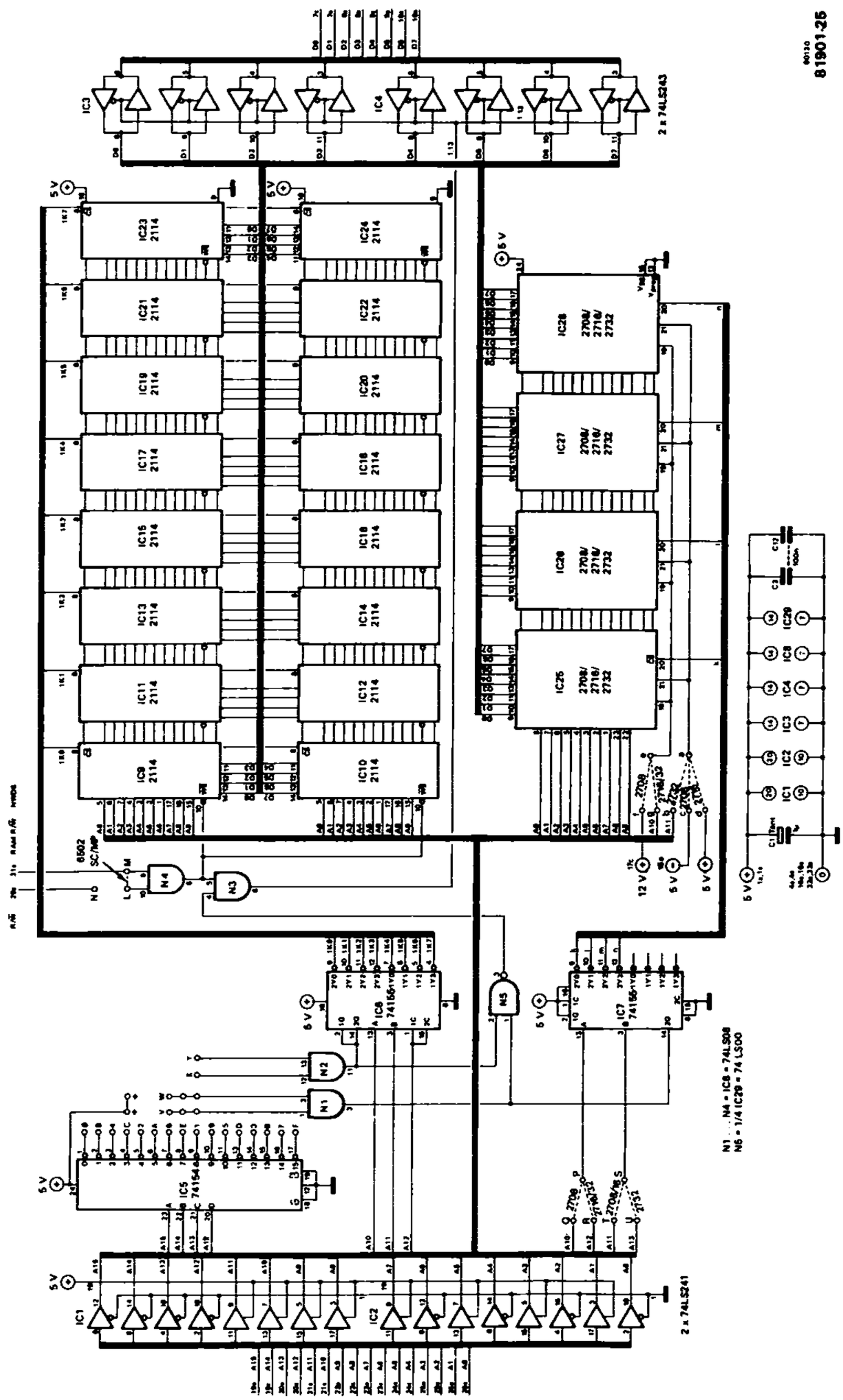


Bild 25. Schaltung der RAM/EPROM-Karte, in diesem Buch auch "Busspeicherkarte" genannt. Für die Lage der Drahtbrücken bei der Bestückung mit EPROMs des Typs 2716 ist im Text dieses Kapitels eine Alternative angegeben.

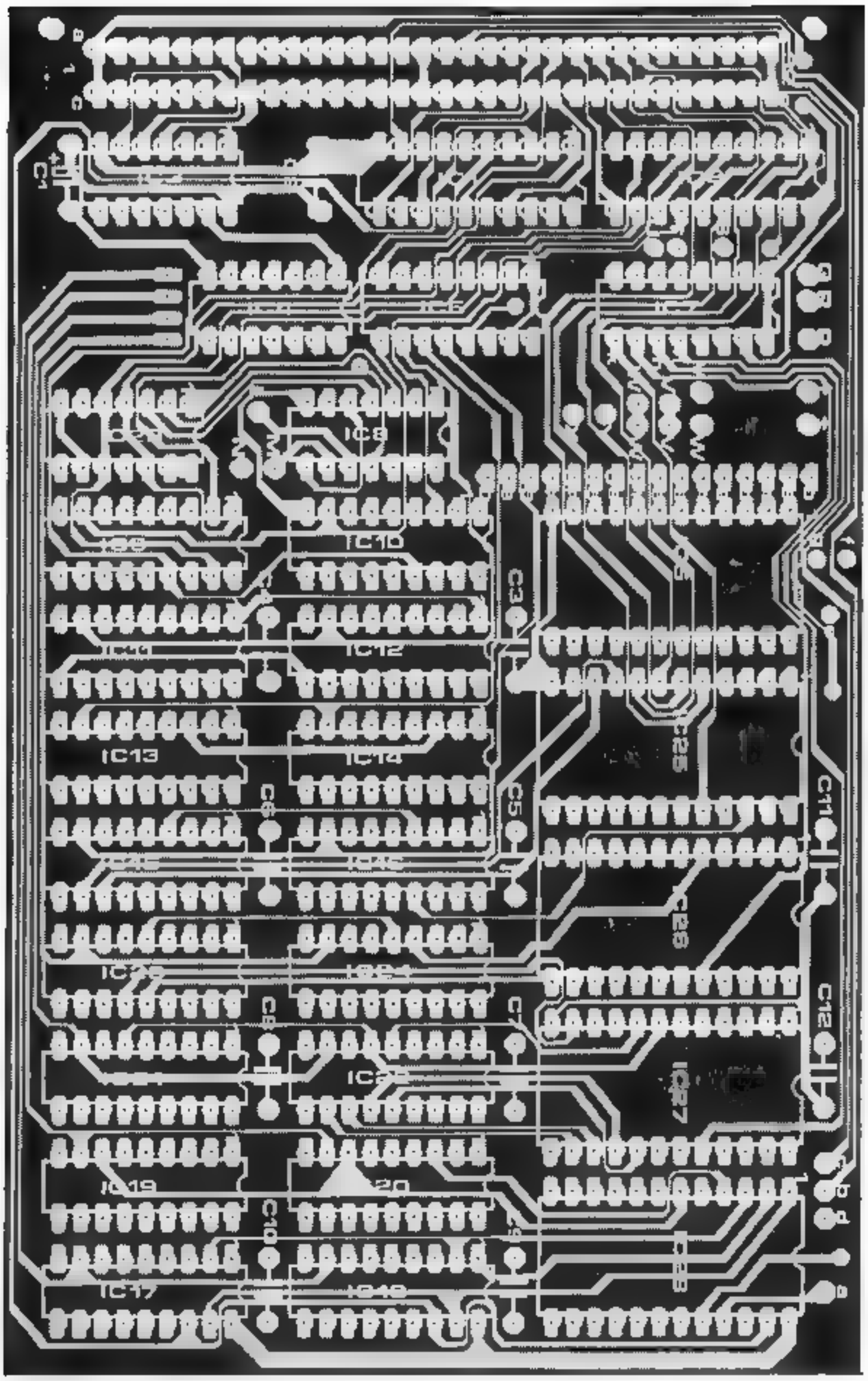
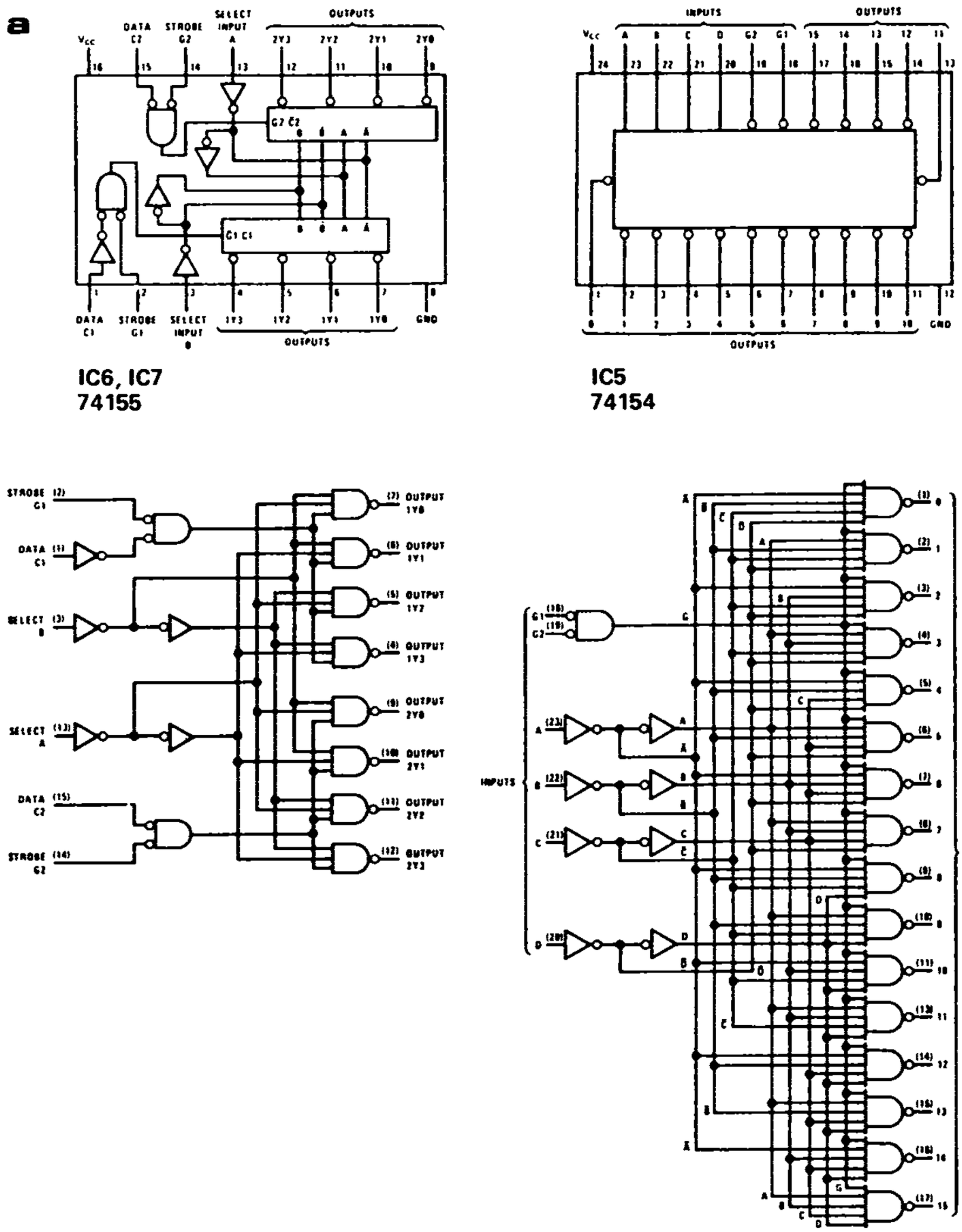


Bild 26. Platine der RAM/EPROM-Karte. Das Bild zeigt in verkleinertem Meßstab die Bestückungsseite und die Leiterbahnen auf der Unterseite.

Tabelle 9. Wahrheitstabelle des Hauptdekoders IC5 auf der RAM/EPROM-Karte.

A15 (A)	A14 (B)	A13 (C)	A12 (D)	Adresse (hex)	welcher Ausgang ist logisch Ø?	
					Platinen-anschluß-punkt	IC5-Pin
Ø	Ø	Ø	Ø	ØXXX	Ø	1
Ø	Ø	Ø	1	1XXX	1	9
Ø	Ø	1	Ø	2XXX	2	5
Ø	Ø	1	1	3XXX	3	14
Ø	1	Ø	Ø	4XXX	4	3
Ø	1	Ø	1	5XXX	5	11
Ø	1	1	Ø	6XXX	6	7
Ø	1	1	1	7XXX	7	16
1	Ø	Ø	Ø	8XXX	8	2
1	Ø	Ø	1	9XXX	9	10
1	Ø	1	Ø	AXXX	A	6
1	Ø	1	1	BXXX	B	15
1	1	Ø	Ø	CXXX	C	4
1	1	Ø	1	DXXX	D	13
1	1	1	Ø	EXXX	E	8
1	1	1	1	FXXX	F	17

miteinander verwechselt werden. Aus Tabelle 9 geht hervor, daß das ganz links stehende halbe Byte der Adresse (ØXXX . . . FXXX) die Nummer des 4-K-Blocks angibt; die übrigen drei halben Byte (XXX) können zwischen ØØØ und FFF variieren. Für die sekundäre Adressendekodierung ist bei den RAMs der Dekoder IC6 zuständig; er wird von drei niedrigeren Bit des Adreßbus und vom Ausgangssignal des Gatters N2 gesteuert. Dekoder IC6 erzeugt acht $\overline{CS}$-Signale für maximal sechzehn RAM-ICs des Typs 2114, so daß die maximale RAM-Kapazität einer Karte 8 K beträgt. Da dies zwei 4-K-Blöcke sind, ist das vom Hauptadressendekoder IC5 gesteuerte Gatter N2 (Anschlüsse X und Y) notwendig. Die sekundäre Adressendekodierung für die EPROMs übernimmt Dekoder IC7 (siehe ebenfalls Bild 27b). Dieser wird abhängig vom EPROM-Typ von zwei niedrigeren Bit des Adreßbus sowie vom Ausgangssignal des Gatters N1 gesteuert, das seine Eingangssignale über die Punkte V und W von Hauptadressendekoder IC5 erhält.
Bei der Belegung der Seiten mit RAMs und EPROMs hat man weitgehend freie Hand. Empfohlen wird für den mit RAM/EPROM-Karten belegbaren 56-K-Bereich (Adressen 2ØØØ . . . FFFF) folgende Einteilung:
RAM: Die Belegung mit RAMs beginnt zweckmäßigerweise bei Adresse 2ØØØ und setzt sich nach höheren Adressen hin fort.
EPROM: Hier wird die umgekehrte Richtung eingeschlagen: Mit Adresse FFFF beginnend setzt sich der mit EPROMs belegte Bereich nach niedrigeren Adressen hin fort.
Anmerkung:
Dieses Konzept gilt unabhängig davon, ob der erweiterte Junior-Computer mit einer oder mehreren Busspeicherkarten bestückt wird. **Achtung! Auf jeden Fall muß Seite FF mit einem EPROM belegt sein, das die Vektoren RST, NMI und IRQ enthält!** Nur wenn der Adressenbereich 2ØØØ . . . FFFF



81901-27a

Bild 27. Anschlußschema und interne Schaltung der Adressendekoder-ICs auf der RAM/EPROM-Karte (a) sowie die zugehörigen Wahrheitstabellen (b).

leer bleibt, können diese Vektoren aus dem EPROM gelesen werden, das sich auf der Basisplatine befindet.
Von vorstehendem Belegungsplan für den Adressenbereich 2000...FFFF ausgehend sollen nun die Einzelheiten der Dekodierung betrachtet werden.

b

IC5

INPUTS						OUTPUTS															
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = High Level, L = Low Level, X = Don't Care

IC6

3-LINE-TO-8-LINE DECODER
OR 1-LINE-TO-8-LINE DEMULTIPLEXER

INPUTS				OUTPUTS							
SELECT			STROBE OR DATA	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
C†	B	A	G‡	2Y0	2Y1	2Y2	2Y3	1Y0	1Y1	1Y2	1Y3
X	X	X	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H
L	H	H	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H
H	H	L	L	H	H	H	H	H	H	L	H
H	H	H	L	H	H	H	H	H	H	H	L

†C = inputs C1 and C2 connected together
‡G = inputs G1 and G2 connected together
H = high level, L = low level, X = don't care

IC7

INPUTS				OUTPUTS			
SELECT		STROBE	DATA	2Y0	2Y1	2Y2	2Y3
B	A	G2	C2				
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

81901-27b

RAM

Die Anzahl der RAM-ICs (2114) muß ein Vielfaches von 2 betragen, da jedes IC 1024 Wörter zu je 4 Bit aufnimmt. In Tabelle 10 ist die empfohlene Adressierung und Dekodierung der einzelnen ICs für eine Belegung der Karte mit 1 . . . 8 K angegeben. Die Drahtbrücken müssen wie folgt gelegt werden:
X – 2 und Y – +5 V (oder Y – X) für 1 . . . 4 K; dekodierte Adressen 2000 . . . 3FFF,
X – 2 und Y – 3 für 5 . . . 8 K; dekodierte Adressen 2000 . . . 3FFF.
Um die RAM-Speicherkapazität auf 9 . . . 16 K auszudehnen, ist eine weitere RAM/EPROM-Karte notwendig. Tabelle 11 gibt die Adressierung und Dekodierung für diese zweite Karte an. Hier sind die Drahtbrücken so zu legen:
X – 4 und Y – +5 V (oder Y – X) für **zusätzliche** 1 . . . 4 K; dekodierte Adressen 4000 . . . 4FFF,
X – 4 und Y – 5 für **zusätzliche** 5 . . . 8 K; dekodierte Adressen 4000 . . . 5FFF.
Für eine noch höhere RAM-Kapazität werden weitere RAM/EPROM-Karten benötigt. Die Punkte X und Y sind über die Brücken wie folgt zu verbinden:

Tabelle 10. Seitenbelegung der RAMs auf der ersten RAM/EPROM-Karte.

Speicher-kapazität	IC9/10	IC11/12	IC13/14	IC15/16	IC17/18	IC19/20	IC21/22	IC23/24
1K	20...23	nein	nein	nein	nein	nein	nein	nein
2K	20...23	24...27	nein	nein	nein	nein	nein	nein
3K	20...23	24...27	28...2B	nein	nein	nein	nein	nein
4K	20...23	24...27	28...2B	2C...2F	nein	nein	nein	nein
5K	20...23	24...27	28...2B	2C...2F	30...33	nein	nein	nein
6K	20...23	24...27	28...2B	2C...2F	30...33	34...37	nein	nein
7K	20...23	24...27	28...2B	2C...2F	30...33	34...37	38...3B	nein
8K	20...23	24...27	28...2B	2C...2F	30...33	34...37	38...3B	3C...3F

Tabelle 11. Seitenbelegung der RAMs auf der zweiten RAM/EPROM-Karte.

Speicher-kapazität	IC9/10	IC11/12	IC13/14	IC15/16	IC17/18	IC19/20	IC21/22	IC23/24	
9K	40...43	nein	nein	nein	nein	nein	nein	nein	**inklusiv** 8K aus Tabelle 10 (erste RAM/EPROM-Karte)
10K	40...43	44...47	nein	nein	nein	nein	nein	nein	
11K	40...43	44...47	48...4B	nein	nein	nein	nein	nein	
12K	40...43	44...47	48...4B	4C...4F	nein	nein	nein	nein	
13K	40...43	44...47	48...4B	4C...4F	50...53	nein	nein	nein	
14K	40...43	44...47	48...4B	4C...4F	50...53	54...57	nein	nein	
15K	40...43	44...47	48...4B	4C...4F	50...53	54...57	58...5B	nein	
16K	40...43	44...47	48...4B	4C...4F	50...53	54...57	58...5B	5C...5F	

dritte Karte: X – 6
Y – 7
vierte Karte: X – 8
Y – 9
und so weiter.

Wie schon erwähnt, wird die mit dynamischen RAMs bestückte 16-K-RAM-Karte hier nicht besprochen. Es ist jedoch möglich, den Adressenbereich ab Adresse 2ØØØ mit dynamischen RAMs oder gemischt mit dynamischen und statischen RAMs zu belegen. Im jedem Fall muß dafür gesorgt werden, daß die 8-K-Blöcke vollständig bestückt sind. Anderenfalls entstehen "Löcher", so daß Teile eines vom Band gelesenen Programms verlorengehen können.

EPROM: 2716

Bestückt man eine RAM/EPROM-Karte mit EPROMs des Typs 2716, so belegen auch die EPROMs dieser Karte einen Bereich von maximal 4 x 2 K = 8 K. Von Hauptdekoder IC5 müssen deshalb ebenfalls zwei 4-K-Blöcke selektiert werden. Dies geschieht mit zwei Drahtbrücken, die die Punkte V und W mit einem oder zwei Punkten der Reihe 2 . . . F verbinden. Sind zwei Brücken notwendig (6 K oder 8 K), dann muß die eine Brücke an einem Anschluß mit gerader Nummer und die andere Brücke an einem Anschluß mit ungerader Nummer liegen.

Wenn die Brücken an den Punkten P und S, wie in der Schaltung (Bild 25) gezeigt, von P nach R und von S nach T gelegt werden, weicht die Reihenfolge der EPROM-Selektion von der IC-Reihenfolge auf der Platine ab. Dies wird aus Tabelle 12a deutlich; das jeweilige EPROM ist bei k . . . n = logisch Ø selektiert (siehe auch Bild 27b). Voraussetzung ist dafür natürlich, daß Hauptadressendekoder IC5 über Gatter N1 den

Tabelle 12. Selektionsfolge der EPROMs IC25 . . . IC28 (EPROM-Typ: 2716) mit den Brücken P – R und S – T (Tabelle 12a) sowie mit den Brücken P – T und S – R (Tabelle 12b).

a

A12 A	A11 B	IC25 k	IC26 l	IC27 m	IC28 n
Ø	Ø	Ø	1	1	1
Ø	1	1	1	Ø	1
1	Ø	1	Ø	1	1
1	1	1	1	1	Ø

b

A12 B	A11 A	IC25 k	IC26 l	IC27 m	IC28 n
Ø	Ø	Ø	1	1	1
Ø	1	1	Ø	1	1
1	Ø	1	1	Ø	1
1	1	1	1	1	Ø

EPROM-Teil der Karte selektiert. Legt man die Brücken nicht von P nach R und von S nach T, sondern statt dessen von P nach T und von R nach S, dann gilt Tabelle 12b. Jetzt stimmt die Reihenfolge der EPROM-Selektion mit der IC-Reihenfolge überein, so daß später ohne langes Überlegen EPROMs in die Fassungen gesteckt werden können, die ein längeres zusammenhängendes Programm enthalten.
Die empfohlene Adressierung und Dekodierung für die EPROMs ist in Tabelle 13 angegeben. Hierzu gehören folgende Drahtbrücken auf der Platine:
e – g und a – d; P – T und R – S;
V – F und W – V für 2 K oder 4 K;
V – F und W – E für 6 K oder 8 K.
Will man mehr als 8 K an EPROMs unterbringen und sieht man von der Bestückung mit EPROMs des Typs 2732 ab, dann ist eine zweite RAM/EPROM-Karte erforderlich. Für die Adressierung und Dekodierung der EPROMs auf der zweiten Karte gilt Tabelle 14; die Brücken auf der Platine sind hier wie folgt zu legen:
e – g und a – d; P – T und R – S;
V – D und W – V für 2 K oder 4 K **zusätzlich** (insgesamt 10 K oder 12 K);
V – D und W – C für 4 K oder 6 K **zusätzlich** (insgesamt 14 K oder 16 K).
Reicht dies noch nicht aus, dann werden weitere Karten hinzugefügt; die Punkte V und W sind absteigend mit dem nächsten bzw. mit den nächsten beiden Anschlußpunkten von Hauptadressendekoder IC5 zu verbinden.

Tabelle 13. Seitenbelegung von 2716-EPROMs auf der ersten RAM/EPROM-Karte.

Speicher-kapazität EPROM	IC25	IC26	IC27	IC28
2K	nein	nein	nein	F8...FF
4K	nein	nein	F0...F7	F8...FF
6K	nein	E8...EF	F0...F7	F8...FF
8K	E0...E7	E8...EF	F0...F7	F8...FF

Tabelle 14. Seitenbelegung von 2716-EPROMs auf der zweiten RAM/EPROM-Karte.

Speicher-kapazität EPROM	IC25	IC26	IC27	IC28	
10K	nein	nein	nein	D8...DF	**inklusiv** 8K aus Tabelle 10 (erste RAM/EPROM-Karte)
12K	nein	nein	D0...D7	D8...DF	
14K	nein	C8...CF	D0...D7	D8...DF	
16K	C0...C7	C8...CF	D0...D7	D8...DF	

EPROM: 2708

Ersetzt man in Tabelle 12a die Adreßleitungen A11 durch A10 und A12 durch A11, dann werden die für die Bestückung mit 2708-EPROMs benötigten Selektionssignale erzeugt. Für die erste RAM/EPROM-Karte sieht

nun die empfohlene EPROM-Adressierung und -Dekodierung wie in Tabelle 15 angegeben aus. Dazu gehören folgende Brücken auf der Platine:
e – f und a – c; P – Q und S – T; V – F und W – V.
Bei mehr als 4 K an EPROMs muß nun bereits die zweite Karte gefüllt werden; für sie gilt Tabelle 16 mit den Brücken
e – f und a – c; P – Q und S – T; V – E und W – V.
Wird die EPROM-Kapazität durch weitere ICs vom Typ 2708 ausgebaut, dann sind mit den Punkten V und W der folgenden Karten die Punkte D, C, B usw. zu verbinden.

Tabelle 15. Seitenbelegung von 2708-EPROMs auf der ersten RAM/EPROM-Karte.

Speicher-kapazität EPROM (2708)	IC25	IC26	IC27	IC28
1K	nein	nein	nein	FC...FF
2K	nein	nein	F8...FB	FC...FF
3K	nein	F4...F7	F8...FB	FC...FF
4K	FØ...F3	F4...F7	F8...FB	FC...FF

Tabelle 16. Seitenbelegung von 2708-EPROMs auf der zweiten RAM/EPROM-Karte.

Speicher-kapazität EPROM (2708)	IC25	IC26	IC27	IC28	
5K	nein	nein	nein	EC...EF	**inklusiv** 4K auf der ersten Karte
6K	nein	nein	E8...EB	EC...EF	
7K	nein	E4...E7	E8...EB	EC...EF	
8K	EØ...E3	E4...E7	E8...EB	EC...EF	

Und schließlich . . .
. . . noch einige wichtige Tips

1. Die Drahtbrücke L – M darf nicht vergessen werden!
2. Die RAM/EPROM-Karten werden mit männlichen Konnektoren (siehe Bild 19a) ausgerüstet; der Konnektor befindet sich auf der Bestückungsseite. Die Buskarte ist mit ein bis fünf weiblichen Konnektoren (Bild 19b) bestückt. Die RAM/EPROM-Karten stehen senkrecht auf der Buskarte.
3. Man beachte, daß die Anschlüsse Ø . . . F von Dekoder IC5 nicht in dieser Reihenfolge, sondern in gleicher Folge wie die zugehörigen IC-Pins auf der Platine angeordnet sind. Dies kann beim Legen der Drahtbrücken zu Irrtümern führen.
4. Ein eventuell freier Anschlußpunkt der Gattereingänge V, W, X und Y wird mit dem zweiten Eingang des gleichen Gatters oder mit +5 V verbunden.
5. Die für die EPROMs empfohlene Seitenbelegung (beginnend mit Adresse FFFF in Richtung niedrigerer Adressen) trägt der Tatsache Rechnung, daß Seite FF mit einem EPROM belegt sein **muß**. Aus dieser Seite

liest der Prozessor nämlich die Vektoren RST, NMI und IRQ. Dies wurde in Zusammenhang mit der Datenbussteuerung ausführlich besprochen. Solange der Monitor benutzt wird, der im EPROM auf der Basiskarte steht, muß das EPROM IC28 der Seite FF folgende Daten enthalten:

Adresse FFFA: 2F (NMIL)
Adresse FFFB: 1F (NMIH)
Adresse FFFC: 1D (RESL)
Adresse FFFD: 1C (RESH)
Adresse FFFE: 32 (IRQL)
Adresse FFFF: 1F (IRQH)

Nach Drücken der Resettaste springt der Prozessor zur Startroutine RESET des Monitors. Die Sprungvektoren NMI und IRQ können auf die bekannte Weise im PIA-RAM (Seite 1A) abgelegt werden. Hier sind auch andere Lösungen möglich, wenn ein anderes Systemprogramm an die Stelle des Monitors tritt. Man beachte, daß der Junior-Computer ohne Spezifizierung des Resetvektors nicht in Gang gesetzt werden kann. Die Vektoren NMI und IRQ sind mit ebenfalls in einem EPROM abgelegten indirekten JMP-Befehlen so zu spezifizieren, daß der Prozessor in einem RAM-Bereich springt. In diesen können dann Vektoren entsprechend den jeweiligen Erfordernissen gesetzt werden. Wem diese Zusammenhänge nicht mehr geläufig sind, der schlage Buch 1, Kapitel 3, Bild 32 auf.

6. EPROMs, die Systemprogramme oder andere Programme enthalten, können im Zuge von Erweiterungen nicht einfach von einem zum anderen IC-Sockel "versetzt" werden. Dann ändert sich nämlich nicht nur die Startadresse, sondern es müssen auch alle Operandendaten (in jedem Fall ADH) angepaßt werden, die absolute Adressen innerhalb des EPROMs betreffen. Wenn ein EPROM seinen Platz wechseln soll, ist dieses EPROM unter Berücksichtigung der geänderten Adressen neu zu programmieren. Dazu muß immer eine ausführliche Programmdokumentation vorhanden sein. Ein Hexdump reicht nicht aus, es sei denn, daß man über ein Disassembler-Systemprogramm verfügt oder das Programm von Hand disassembliert. Letzteres ist jedoch mit erhöhter Fehlergefahr verbunden.

7. Ohne weiteres kann eine RAM/EPROM-Karte mit EPROMs vom Typ 2716 bestückt werden, während man auf eine andere 2708-EPROMs setzt. Die Belegung von Adresse FFFF an abwärts wird beibehalten; die Lage der Drahtbrücken muß entsprechend der Bestückung gewählt werden. Im EPROM-Bereich sind eventuelle "Löcher" weniger gefährlich als im RAM-Bereich, insbesondere wenn der 56-K-Erweiterungsbereich nur zu einem kleinen Teil belegt wird.

8. Es ist zulässig, einen von IC5 dekodierten 4-K-Block auf einer RAM/EPROM-Karte nur teilweise mit RAMs bzw. EPROMs zu bestücken. **Unzulässig ist jedoch, den auf dieser Karte nicht bestückten Teilbereich auf einer anderen Karte zu belegen.** In diesem Fall werden nämlich gelesene Daten verfälscht, weil die Ausgänge von jeweils zwei Datenbuspuffern parallelgeschaltet sind (vergleiche Bild 6).
Schlußfolgerung: Jeder Anschlußpunkt Ø . . . F von IC5 darf auf sämtlichen RAM/EPROM-Karten insgesamt nur ein einziges Mal benutzt werden.

10. Eine einzelne RAM/EPROM-Karte läßt sich auch unmittelbar an die Basiskarte anschließen. Näheres geht aus dem Anhang 1 am Schluß des Buchs hervor.

Das 10. Kapitel ist damit abgeschlossen. Im nächsten Kapitel wenden wir uns dem Anschluß der Kassettenrekorder, dem Einstellen des PLL und dem praktischen Umgang mit dem Systemprogramm TAPE MONITOR zu. Wer sich zunächst nur theoretisch orientieren will, wird vor dem Umsetzen des Gelernten in die Praxis noch das 12. Kapitel studieren. Dort werden die anschließbaren peripheren Komponenten sowie das Systemprogramm PRINTER MONITOR behandelt. Ob Praxis oder zunächst nur Theorie, das zurückliegende 10. Kapitel war erst der Anfang einer neuen Entdeckungsreise in die faszinierende Welt der Computertechnik.

11

Magnetische Speichererweiterung

Daten vom, zum und am laufenden Band

Die Hardware des Kassetteninterface und die Software des Systemprogramms Tape Management (TM) arbeiten Hand in Hand, wenn Daten vom Junior-Computer zur Tonbandkassette oder in umgekehrter Richtung zu transportieren sind. Dieser Vorgang hat Ähnlichkeit mit dem, was der Prozessor bei den Befehlen LDA und STA ausführt. Nun werden jedoch nicht nur einzelne Bytes, sondern ganze Datenblöcke transportiert. Ein zweiter Unterschied ist noch bedeutsamer: Im Gegensatz zum RAM, das beim Abschalten der Speisespannung seinen Inhalt verliert, bleiben Daten, zum Beispiel Programme, auf dem Magnetband beliebig lange stehen. Damit ist der Weg zum Aufbau eines Programmarchivs und einer Programmbibliothek geebnet, ohne daß viel zeitraubende und mühsame Tipparbeit geleistet werden muß.
Wie läuft der Datentransport ab? Was ist beim Einstellen des PLL zu beachten? Welche Tasten sind zu drücken, um Daten auf Band zu speichern und vom Band zu lesen? Diese und andere Fragen werden nachfolgend so ausführlich beantwortet, daß, so hoffen wir, sich das Band zum Junior-Computer um ein weiteres Stück festigt.

Das Speichern von umfangreichen Datenblöcken auf Magnetband dient nicht nur der Bequemlichkeit, sondern ist ein einfaches und vor allem kostengünstiges Verfahren, um das "vergeßliche" interne Gedächtnis des Junior-Computers zu entlasten. Wenn wir beispielsweise ein Programm entwickelt haben und dieses erst später laufen lassen wollen, so wird dieses Programm einfach in das magnetische "Notizbuch" geschrieben. Das RAM des Juniors ist dann wieder frei für neue Aktivitäten. Oder wir schalten den Computer aus: was der Junior "vergißt", bewahrt die Kassette auf. Später liest der Computer dann selbst vom Band, was ihm inzwischen entfallen ist. Wir brauchen ihm nicht durch mühsames und fehlerträchtiges Eintippen zu seinem früheren Wissensstand zurückverhelfen.
Oder nehmen wir den Fall, daß wir ein umfangreiches Programm editieren. Wir können das Programm in editierter, also noch nicht assemblierter Form auf Band setzen. Dann folgt unabhängig von der Bandkonserve das Assemblieren und der Probelauf des Programms. Dabei zeigt sich aller Erfahrung nach, daß das Programm noch unvollkommen ist oder sogar überhaupt nicht läuft. Nun muß das Programm nicht noch einmal die Prozedur des Editierens durchlaufen (nach dem Assemblieren sind bekanntlich die Label verschwunden). Wir lesen das editierte Programm wieder ein, starten den Editor warm und korrigieren die zuvor unterlaufenen Fehler. Vielleicht kommen dabei noch einige Label hinzu. Auch jetzt können wir wieder das Programm, in der geänderten Version, vor dem Assemblieren auf Band setzen. Das Ganze läßt sich so oft wiederholen, bis alle Fehler beseitigt sind. Die Bandkassette hat dabei die Funktion einer Datenbank: aufbewahrt wird nicht unser Geld, sondern unser Datenblock.
Doch das ist nur eine von vielen Möglichkeiten. Eine andere ist die Nutzung einer Programmbibliothek. Dies ist eine Sammlung von Programmen, die wir nicht selbst entwickelt haben, jedoch ausleihen oder kaufen können. Oft werden solche Programmbibliotheken von Hobbycomputer-Clubs eingerichtet. Der Austausch von Software mit einem anderen Junior-Computer-Besitzer ist durch die Bandkassette ebenfalls leicht möglich. Natürlich können wir fremde Programme auch Byte für Byte oder Instruktion für Instruktion eintippen, wenn das Programm in geschriebener oder gedruckter Form vorliegt. Bei längeren Programmen wirkt sich diese rein handwerkliche Tätigkeit jedoch nicht nur sehr ermüdend aus, sondern die Gefahr, eine falsche Taste zu drücken, ist außerordentlich groß. Der Mensch erweist sich auch hier als Fehlerquelle ersten Ranges; das Schreiben und Lesen des Programms von Band ist unvergleichlich zuverlässiger.

Die Bandkassette

ein magnetisches RAM

Lohnend ist die Investition von rund hundert Mark für einen Mono-Kassettenrekorder mit Bandzählwerk auf jeden Fall, oder ist ein derartiges Gerät etwa schon vorhanden? Dann fehlen zum Aufbau eines Programmarchivs nur noch einige C-60-Kassetten. Eine solche Kassette hat bekanntlich pro Durchlauf eine Spieldauer von 30 Minuten. Hiervon müssen die Start- und Stoppzeichen sowie die Pausen abgezogen werden, so daß für die Pro-

grammaufzeichnung rund 25 Minuten übrig bleiben. Bei einer Geschwindigkeit von 50 Datenbyte pro Sekunde (wie dieser Wert zustande kommt, wird später erklärt) lassen sich auf jeder Spur 25 mal 60 mal 50 Byte speichern; dies sind mehr als 73 KByte (1 KByte = 1024 Bytes)! Vergleicht man Kapazität und Kosten einer C-60-Bandkassette mit anderen Datenträgern, dann erweist sich nur ein 100-Blatt-Notizblock im Format DIN A4 als preiswerter.

Daten auf Magnetband

Wir wollen uns nun mit dem "Rahmen" beschäftigen, der jeden Datenblock begleitet. Der Rahmen dient unterschiedlichen Zwecken: der Unterscheidung des Datenblocks von nicht für den Computer bestimmten Informationen, der Identifizierung eines Datenblocks unter mehreren anderen Datenblöcken und der Kontrolle auf fehlerfreie Datenübermittlung. Wir unterscheiden deshalb die

- **Datensendung** und den
- **Datenblock.**

Zu jeder Datensendung gehören die eigentlichen Daten sowie eine "Einleitung" und ein "Schlußakkord". In Bild 1 ist skizziert, in welcher Reihenfolge die Glieder einer Datensendung auf dem Band stehen. Die Datensendung hat Ähnlichkeit mit einer Schlange: An den Kopf schließt sich ein mehr oder weniger langer Körper (der Datenblock) an, gefolgt von einem Schwanz. Lassen wir nun die Körperteile der Schlange am Tonkopf des Kassettenrekorders vorüberziehen:

1 255 Synchronisationszeichen

Diese dienen zur eindeutigen Unterscheidung vom eventuell vorangesetzten, gesprochenen Kommentar ("Dies ist meine erste 24-Stunden-Software-

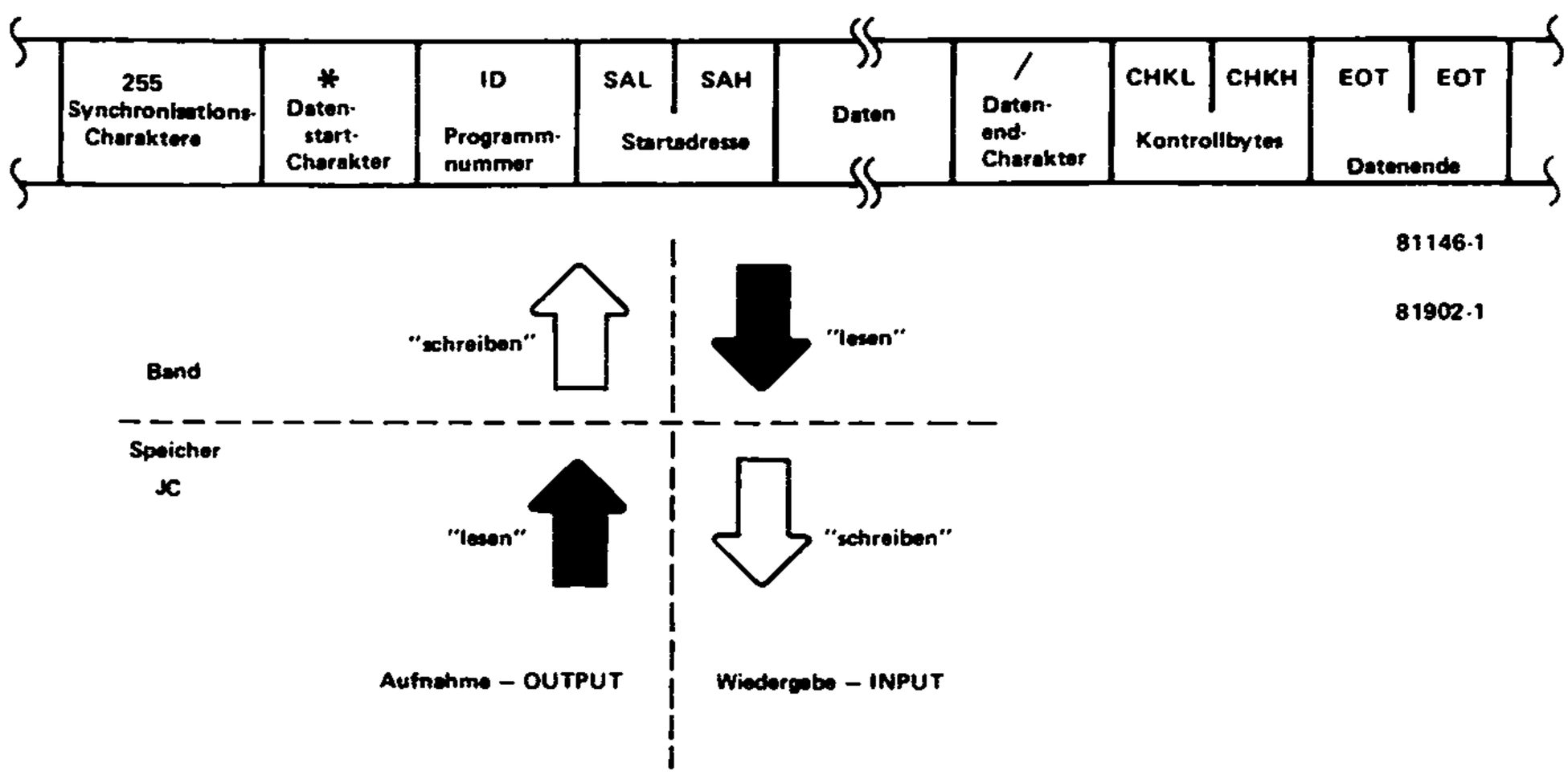


Bild 1. So stehen die Daten auf dem Band: Der Datenblock wird von einem "Vorspann' und einem "Nachspann" begleitet.

Uhr" usw.). Beim Lesen vom Band muß der Computer die nicht für ihn bestimmten Informationen unbeachtet lassen; seine Hardware darf daraus zwar logische Nullen und Einsen produzieren, sie dürfen jedoch nicht als Daten in das RAM gesetzt werden. Alle zu einer Datensendung gehörenden Zeichen stehen in sogenannten ASCII-Format auf dem Band, einem Sieben-Bit-Kode, zu dem noch ein achtes Bit hinzugefügt ist. Die sieben Bit repräsentieren die Buchstaben, Ziffern und Satzzeichen der uns geläufigen "Klarschrift" sowie noch einige Sonderzeichen. Das achte Bit dient speziellen Zwecken; hier ist es stets Ø. Eine ASCII-Tabelle ist im Anhang 7 am Schluß dieses Buches zu finden. Die einzelnen Bit der ASCII-Bytes sind **nacheinander** (seriell) auf dem Band untergebracht; in Kapitel 12 wird die serielle Datenübertragung noch ausführlicher besprochen. Die Synchronisationszeichen haben in hexadezimaler Schreibweise den ASCII-Kode 16. Am Anfang jeder Datensendung steht also 255 mal die Bitfolge ØØØ1Ø11Ø.

2 Datenblockbeginnzeichen*

Es signalisiert, daß die Synchronisationszeichen beendet sind und nun der Datenblock folgt. Dieses Zeichen hat den hexadezimalen ASCII-Kode 2A, der für * (Stern) steht.

3 Programmnummer ID

Damit sich gespeicherte Datenblöcke, zum Beispiel Programme, auf dem Band leicht wiederfinden lassen, werden sie fortlaufend numeriert: ID bedeutet "Identifikation". Maximal können 254 Programmnummern vergeben werden, das entspricht den hexadezimalen Zahlen Ø1 . . . FE. Die Zahlen ØØ und FF sind ausgenommen; sie haben eine spezielle Bedeutung, die später noch zur Sprache kommt. Die gewählte Programmnummer wird vor dem Schreiben eines Datenblocks über das hexadezimale Tastenfeld in den Computer eingegeben. Vor dem Lesen eines Datenblocks wird dem Computer auf die gleiche Weise mitgeteilt, welchen Datenblock er vom Band in das RAM transportieren soll.

4 Rechtes Startadreßbyte SAL

5 Linkes Startadreßbyte SAH

und ferner die letzte Adresse LA und die Endadresse EA

Die Bedeutung dieser Adressen ist in Bild 2 dargestellt. Startadresse SA ist die erste Adresse des zu schreibenden oder zu lesenden Datenblocks. Bevor Daten auf das Band gesetzt werden können, muß der Junior-Computer natürlich wissen, an welcher Stelle im RAM er damit beginnen soll. Dies ist die Startadresse SA. Die letzte Adresse (LA) ist die Adresse, deren Inhalt als letztes auf das Band gesetzt wird. Nicht diese Adresse muß dem Computer vor dem Datentransport zum Band mitgeteilt werden, sondern die um einen Schritt höhere Endadresse EA. Aus Bild 1 geht hervor, daß Endadresse EA nicht auf dem Band steht. EA ergibt sich automatisch aus der Startadresse SA und der Länge des Datenblocks, so daß beim Lesen nur die Programmnummer ID und in bestimmten Fällen auch die Startadresse SA eingegeben werden muß.

Die Unterscheidung der letzten Adresse LA und der um einen Schritt höheren Endadresse EA steht übrigens in Zusammenhang mit der Software-Kompatibilität zum Mikrocomputer KIM. Ferner noch folgender Hinweis, der vielleicht überflüssig ist: Die Startadresse SA muß stets niedriger als die Endadresse EA sein, denn alle Daten werden in der Reihenfolge

aufsteigender Adressen auf das Band gesetzt. Ist andererseits die Startadresse SA höher oder gleich der Endadresse EA, überträgt der Junior-Computer nur die Synchronisationszeichen und unterbricht dann die Aufzeichnung.
Weshalb ist die Speicherung der Startadresse SA auf dem Band notwendig? Es wäre doch bequemer, die Startadresse erst vor dem Beginn des Lesens frei zu wählen. Das ist zwar möglich, wenn man als Programmnummer FF eingibt (darüber später mehr), doch im allgemeinen entstehen dadurch Komplikationen. Die meisten Programme (= Datenblöcke) enthalten nämlich absolute Adreßoperanden, die sämtlich oder teilweise geändert werden müssen, wenn sich die Startadresse ändert.

6 Der Datenblock

Jedes Datenbyte wurde bei der Aufzeichnung in zwei ASCII-Bytes umgewandelt (pro Nibble ein ASCII-Byte), so daß 16 Bit auf dem Band ein Datenbyte verkörpern. Das erste Datenbyte ist das Byte, daß unter der Startadresse SA im Speicher stand, während das letzte Byte des Datenblocks unter der um eins herabgesetzten Endadresse EA zu finden war (siehe Bild 2). Die Länge eines Datenblocks ist im Prinzip unbegrenzt. Es können auch Daten auf das Band gesetzt werden, die nicht in einem RAM, sondern in einem (E)PROM stehen. In umgekehrter Richtung ist natürlich nur der Datentransport zum RAM möglich, weil das Schreiben von Daten in ein (E)PROM gleichbedeutend mit seiner Programmierung ist.

7 Datenblockendzeichen "/"

Dieses Zeichen signalisiert dem Computer das Ende des Datenblocks. Beim Schreiben von Daten auf Band stimmt hier der Adreßpointer (POINTH, POINTL) mit der Endadresse EA überein. Das Datenblockzeichen hat den hexadezimalen ASCII-Kode 2F.

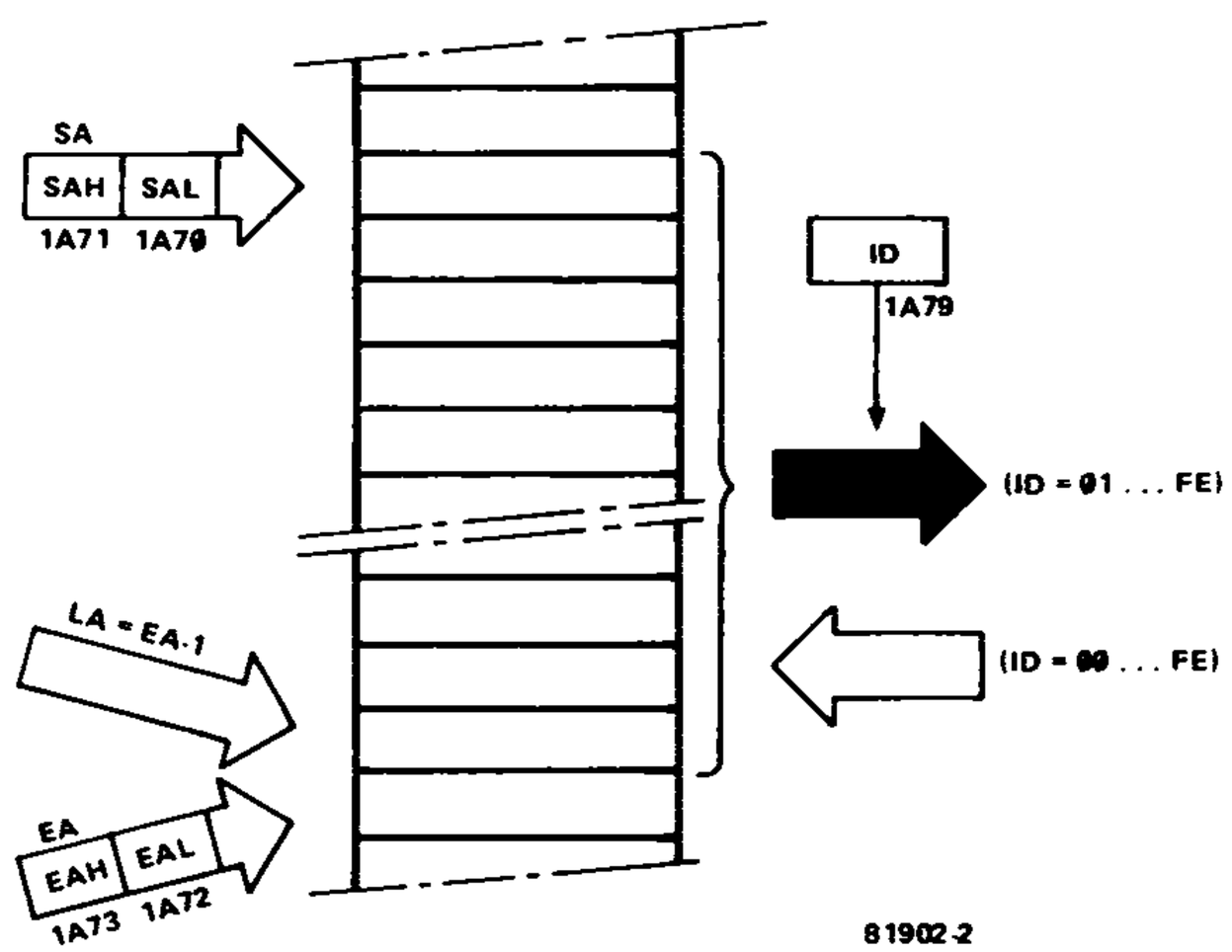


Bild 2. Zu jedem Datenblock gehören eine Startadresse SA, eine Endadresse EA und eine Programmnummer ID. Diese und andere Kennzeichen werden "Parameter" genannt.

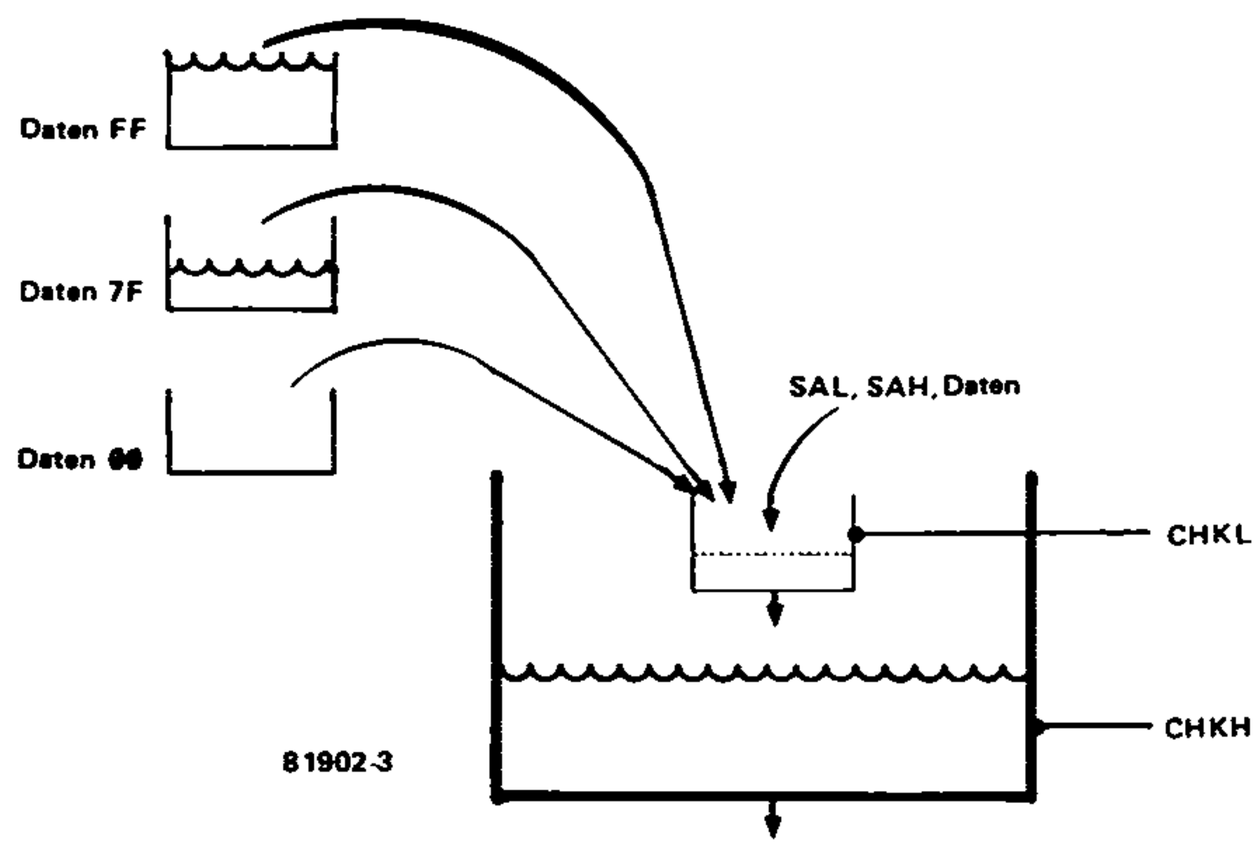


Bild 3. Die Flüssigkeitsbehälter verdeutlichen, wie der Computer die Werte der Kontrollbytes CHKL und CHKH berechnet.

8 Die Kontrollbytes CHKL und CHKH

Mit Hilfe dieser beiden Bytes läßt sich kontrollieren, ob der Datentransport fehlerfrei verlief. Auf dem Weg zur Kassette und später in umgekehrter Richtung kann nämlich unter Umständen einiges schiefgehen. Ursache sind meistens sogenannte Dropouts des Magnetbands; sehr viel seltener führt ein falsch eingestellter PLL (siehe "PLL-Abgleich") zur Verfälschung von Daten. Kontrolliert wird hier wie folgt: Zu Beginn der Aufzeichnung sind die Bytes CHKL und CHKH beide ØØ. Von SAL an werden nacheinander SAL, SAH und sämtliche Daten zu CHKL hinzuaddiert. Dabei handelt es sich um den Zahlenwert der Daten vor ihrer Umwandlung in den ASCII-Kode. Unberücksichtigt bleibt bei der Addition die Programmnummer ID. Die Anzahl der Überläufe von CHKL über den Wert FF wird in CHKH festgehalten. Erreicht auch der Inhalt von CHKH den Wert FF, so beginnt das Ganze von vorn.

Beim Lesen der Daten vom Band läuft genau die gleiche Prozedur ab: Sämtliche Zahlenwerte werden addiert und schließlich mit den auf dem Band stehenden Werten von CHKL und CHKH verglichen. Stimmen sie nicht überein, dann ist beim Datentransport mit hoher Wahrscheinlichkeit ein Fehler unterlaufen. Bei gleichen CHKL- und CHKH-Werten ist dagegen ein fehlerfreier Datenaustausch so gut wie sicher.

Zur Verdeutlichung des Verfahrens betrachten wir eine Parallele aus dem Bereich der Mechanik. In Bild 3 ist ein kleines Gefäß skizziert, das das Fassungsvermögen von CHKL besitzt. Darunter befindet sich ein größeres Gefäß mit dem 256-fachen Inhalt; dies entspricht dem Kontrollbyte CHKH. Der "Datenstrom" fließt beim Schreiben auf Band in das kleinere Gefäß. Immer dann, wenn dieses Gefäß gerade bis zum Rand gefüllt ist, öffnet sich sein Boden. Der Inhalt fällt in das größere Gefäß hinab. Ist auch dieses gerade bis zum Rand gefüllt, so entleert es sich ebenfalls durch kurzzeitiges Öffnen des Bodens. Nach dem Versiegen des Datenstroms ist jedes Gefäß mit einer bestimmten Menge Flüssigkeit gefüllt.

Beim Lesen der Daten vom Band geschieht genau das Gleiche. Anschlie-

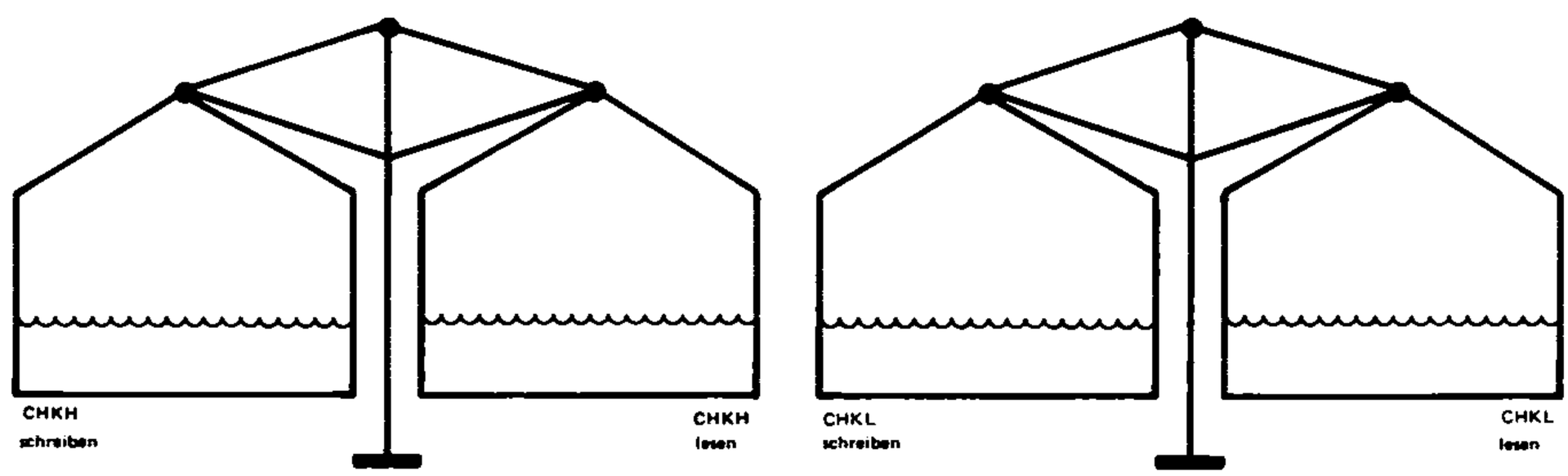


Bild 4. Nach dem Lesen werden die errechneten CHKL- und CHKH-Werte mit den ursprünglichen Werten verglichen. Das Bild zeigt dies am Modell der Flüssigkeitsbehälter.

ßend wird der Flüssigkeitsstand am Ende der Leseprozedur mit den Werten verglichen, die vorher beim Schreiben erreicht wurden. Schlägt der Zeiger der Waage in Bild 4 nach der einen oder der anderen Seite aus, dann ist beim Datentransport höchstwahrscheinlich ein Fehler unterlaufen. Sind dagegen die Flüssigkeitsmengen jeweils gleich groß, so darf mit fast absoluter Sicherheit angenommen werden, daß alles ordnungsgemäß verlief.
Um diese Methode der Datensicherung erschöpfend zu erklären, müßte ein großer Teil dieses Buchs mit Abhandlungen über Statistik, Informationstheorie und ähnlicher trockener Materie gefüllt werden. Aus naheliegenden Gründen wird jedoch darauf verzichtet. Erwähnenswert ist aber, daß die anscheinend unnötige Umwandlung der Datenbytes in jeweils zwei ASCII-Bytes mit der Datensicherung in Zusammenhang steht.
Eine ungefähre Vorstellung von der Grundlage der Methode gibt eine andere Parallele: Eine Sparkasse oder Bank ist bestrebt, das Vermögen ihrer Sparer so sicher wie möglich aufzubewahren. Es muß verhindert werden, daß sich jemand im Gebäude einschließen läßt, um später mit der Beute auszubrechen. Man zählt deshalb die Kunden, die während der Öffnungszeiten die Bank betreten, sowie die Kunden, die die Bank verlassen. Beim Schließen der Bank werden beide Zahlen miteinander verglichen. Wie hoch die Sicherheit ist, daß sich bei Übereinstimmung beider Zahlen niemand mehr im Gebäude aufhält, das hängt von der Zuverlässigkeit der Zählung ab. Es können zum Beispiel zwei Personen gleichzeitig, nebeneinander gehend, die Lichtschranke passieren, oder ein Lottokönig verläßt die Bank durch den Hinterausgang, nachdem er von der Direktion über die Anlagemöglichkeiten seines Reichtums beraten wurde. Schlußfolgerung ist, daß die Methode des Kontrollierens das Maß der Sicherheit bestimmt. Ferner können sich zwei Fehler mit unterschiedlichen Vorzeichen gegenseitig aufheben, so daß nur scheinbar alles in Ordnung ist.
Zurück zu den Kontrollbytes CHKL und CHKH: Diese Art der Datensicherung kann im Gegensatz zur beschriebenen Zählung der Bankkunden als nahezu hundertprozentig zuverlässig betrachtet werden.

9 Zwei Datensende-Endzeichen EOT
Das Erscheinen dieser beiden Zeichen signalisiert unzweideutig das Ende einer Datensendung; EOT bedeutet "end of transmission". Der hexadezimale ASCII-Kode von EOT ist Ø4.
Damit sind wir am Ende der Schlange in Bild 1 angelangt. Bis auf die Anzahl der Synchronisationszeichen (255 anstelle von 100) und die beim Junior-Computer sechsfach höhere Übertragungsgeschwindigkeit, verwendet der Mikrocomputer KIM dasgleiche Datenaufzeichnungsformat.

Wie stehen die Daten auf dem Band?

Digitalmusik

Wir wissen inzwischen, daß die Daten seriell, also Bit für Bit auf dem Band stehen. Die Frage ist jedoch: In welcher Gestalt sind dort logische Nullen und Einsen gespeichert? Da im Computer eine logische 1 durch eine hohe Spannung und eine logische Ø durch eine niedrige Spannung dargestellt wird, der Kassettenrekorder jedoch nur tonfrequente Signale verarbeiten kann, liegt die Lösung nicht allzu fern: Die Einsen und Nullen müssen in hörbare Töne "übersetzt" werden. Genau dies bewerkstelligt das Kassetteninterface zusammen mit der Kassettensoftware. Nun könnte man meinen, daß die Übersetzung einer logischen 1 in einen hohen Ton und einer logischen Ø in einen tiefen Ton ausreicht. Dies hat sich jedoch als nicht genügend zuverlässig erwiesen, wenn man mit hohen Übertragungsgeschwindigkeiten arbeiten will.
Sowohl eine logische Ø als auch eine logische 1 wird deshalb vom Junior-Computer in einen hohen Ton gefolgt von einem tiefen Ton übersetzt. Die beiden unmittelbar aufeinanderfolgenden Töne sind zeitlich jedoch nicht gleich lang; nur die Gesamtdauer beider Töne ist konstant. Bei einer logischen Ø hat der hohe Ton die doppelte Länge des nachfolgenden tiefen Tons, bei einer logischen 1 verhalten sich die Zeiten genau umgekehrt: der hohe Ton hat die halbe Länge des nachfolgenden tiefen Tons.
Den logischen Signalen Ø und 1 entsprechen also nicht zwei Tonsignale mit unterschiedlichen Frequenzen, sondern die Information steckt im Zeitverhältnis von zwei unterschiedlich langen Tonsignalen. In Bild 5a sind die Tonsignale als Noten dargestellt. Bei der "Bit-Ø-Melodie" folgt auf zwei hohe Noten eine tiefe Note, während die "Bit-1-Melodie" aus einer hohen Note gefolgt von zwei tiefen Noten besteht.

Nach diesem einleitenden musikalischen Exkurs wollen wir nun das Verfahren etwas genauer betrachten. In Bild 6a, das wir bereits aus dem vorangegangenen Kapitel kennen, sind die Tonsignale in Form von Impulsreihen gezeichnet. Jede Impulsreihe besteht aus mehreren schmalen Impulsen (hohe Frequenz) und mehreren breiten Impulsen (niedrige Frequenz). Einer logischen 1 entsprechen drei halbe Perioden eines 3600-Hz-Signals gefolgt von vier halben Perioden eines 2400-Hz-Signals (② in Bild 6a). Eine logische Ø ist identisch mit sechs halben Perioden des 3600-Hz-Signals und zwei halben Perioden des 2400-Hz-Signals (① in Bild 6a). Die Impulsreihen für logisch Ø und logisch 1 sind gleich lang; sie haben stets die

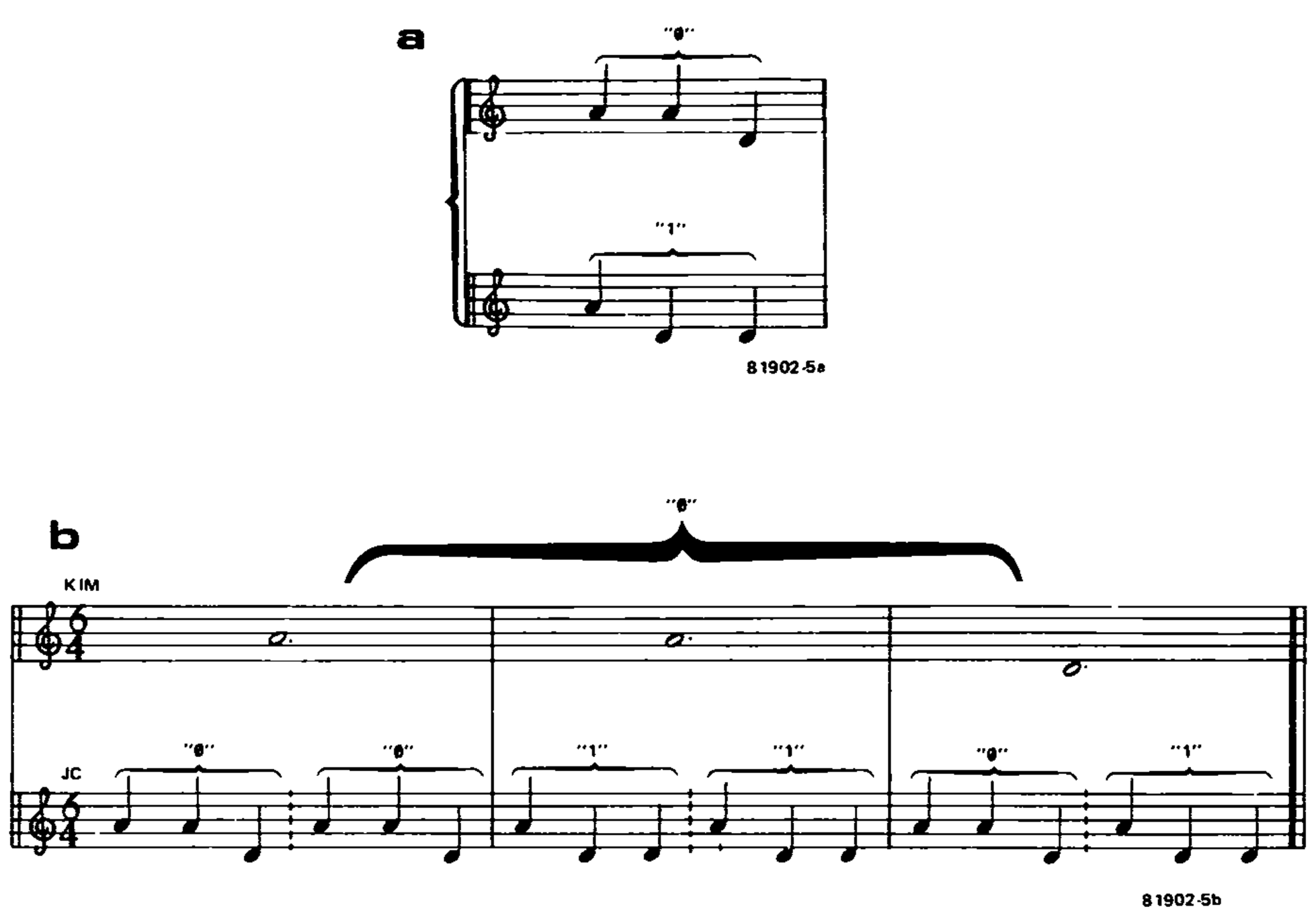
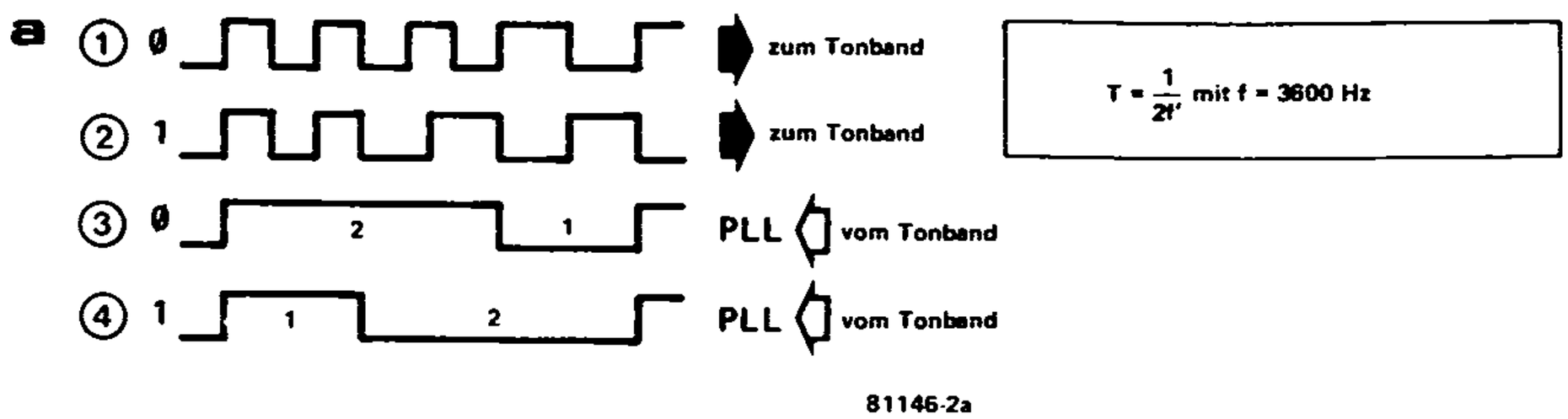


Bild 5. Musikalisch gesehen besteht jedes Bit aus zwei hohen Tönen und einem tiefen Ton oder einem hohen und zwei tiefen Tönen (5a). Der Rhythmus dieser "Bit-Musik" ist beim Junior fast sechsmal so schnell wie beim Standard-KIM (5b).

Dauer 9T, wenn T die halbe Periodendauer des 3600-Hz-Signals ist. Jede Impulsreihe beginnt mit dem höherfrequenten Signal, und außerdem verhält sich die Dauer des höherfrequenten Signals zur Dauer des niederfrequenten Signals entweder wie eins zu zwei oder wie zwei zu eins. Dieser Sachverhalt wurde bereits in Bild 5a in musikalischer Schreibweise dargestellt.
Bild 6b macht zum Vergleich das Verfahren deutlich, das der Mikrocomputer KIM benutzt. Die Zeitachse mußte in drei Abschnitte unterteilt werden, damit die Impulsdiagramme nicht nach rechts über den Rand der Buchseite hinauslaufen. Eine logische 1 besteht hier nämlich aus neun ganzen Perioden eines 3700-Hz-Signals (in Bild 6b auf 3600 Hz abgerundet) und zwölf ganzen Perioden eines 2400-Hz-Signals. Zu logisch Ø gehören achtzehn ganze Perioden des 3700-Hz-Signals und sechs ganze Perioden des 2400-Hz-Signals. Die Dauer jeder Impulsreihe (die Bitlänge) beträgt beim Standard-KIM, verglichen mit dem Junior-Computer, das sechsfache. Auch in der musikalischen Fassung (Bild 5b) kommt die vergleichsweise langsame Übertragungsgeschwindigkeit des KIM mehr als deutlich zum Ausdruck.
Das niedrige Schreib- und Lesetempo des KIM gab verschiedentlich Anlaß, hier nach neuen Wegen zu suchen. Entscheidend trug dazu J. Butterfield mit seiner "Hypertape"-Software bei. In Buch 4, das sich unter anderem



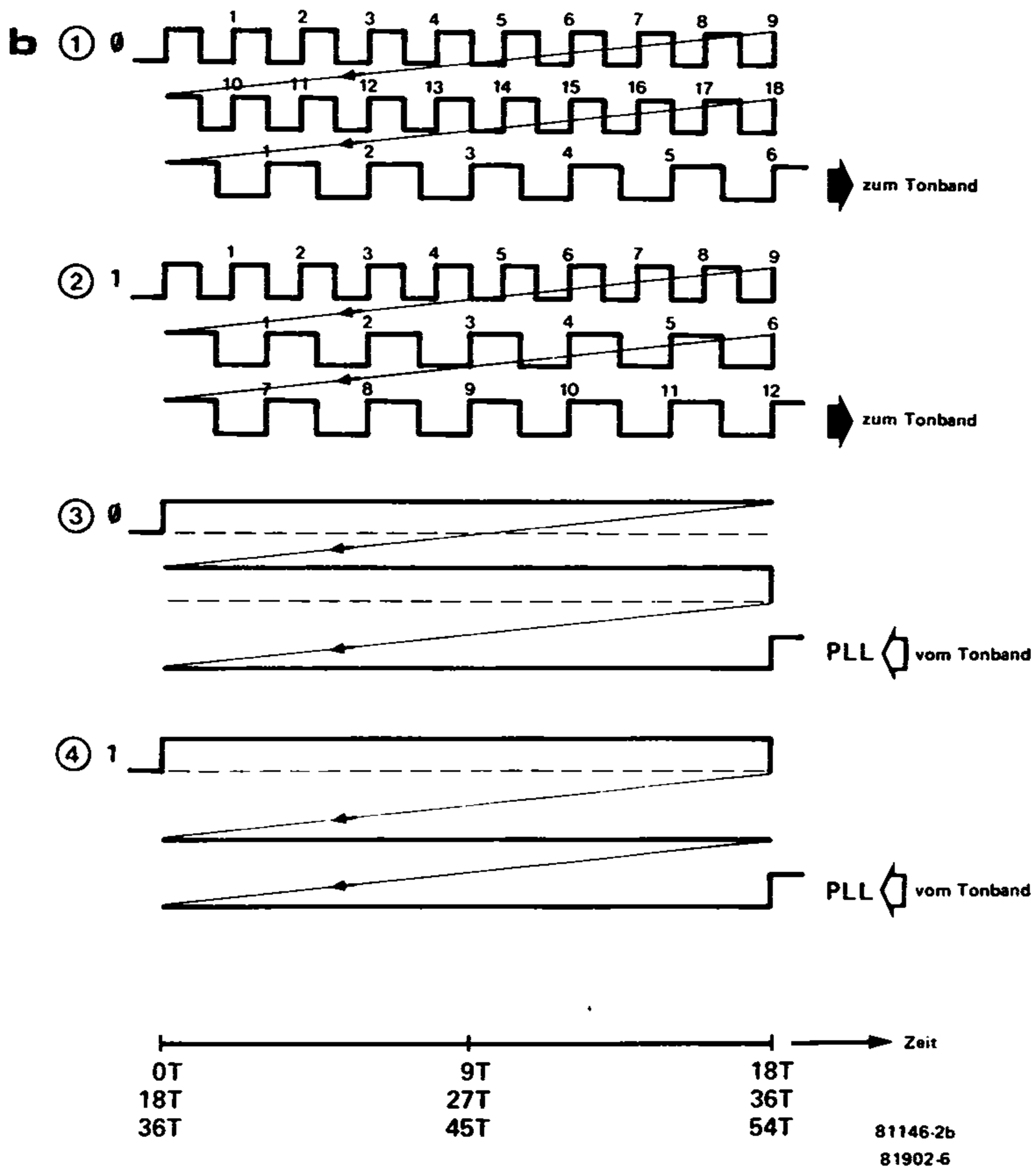


Bild 6. Die Nullen und Einsen werden in tonfrequente Impulsreihen "übersetzt", bevor sie zum Kassettenrekorder gelangen (① und ②). Beim Lesen vom Band erzeugt eine PLL-Schaltung aus den Tönen die Signale ③ und ④. Bild 6a zeigt die Signale des Junior-Computers, Bild 6b gibt die Verhältnisse beim KIM wieder. Die Diagramme des KIM mußten unterteilt werden, weil seine Datenübertragungsgeschwindigkeit im Vergleich zum Junior nur ungefähr ein Sechstel beträgt.

mit der Kassettensoftware des Junior-Computers befaßt, werden die Zusammenhänge zwischen der Übertragungsgeschwindigkeit und der Zuverlässigkeit der Übertragung noch eingehend untersucht.
Wie schon festgestellt, beträgt die Bitlänge der Junior-Kassettensoftware neun halbe Perioden des 3600-Hz-Signals, was 9 · 139 µs = 1250 µs entspricht. Dies ist gleichbedeutend mit 800 bit/s oder 100 ASCII-Byte pro Sekunde oder 50 Datenbyte pro Sekunde. Die Bitgeschwindigkeit wird auch Baudrate genannt; der Junior-Computer arbeitet daher mit einer Baudrate von 800 bit/s.
Auf das Band geschrieben werden die Daten mit der Subroutine DUMP/DUMPT, die vom Hauptprogramm TAPE MANAGEMENT aufgerufen wird. Während des Schreibens ist das sechsstellige LED-Display auf der Basisplatine abgeschaltet; statt dessen leuchtet die rote Aufnahme-LED auf. Bevor Daten auf Band gesetzt werden können, benötigt der Computer noch folgende Informationen: 1. die Programmnummer ID (Ø1 . . . FE), 2. die Startadresse SA und 3. die Endadresse EA, die um eine höher als die letzte Adresse LA des zu übertragenden Datenblocks ist. Von der Startadresse SA bis zur letzten Adresse LA wird der dort stehende Datenblock auf das Band kopiert. Das geschieht mit Hilfe des bereits bekannten Adreßpointers POINT, der auch bei der Anzeige von Adressen und Daten auf dem Display eine Rolle spielt. Unmittelbar nach dem Ende der Datenübertragungsroutine DUMP/DUMPT ist der Inhalt von POINT gleich EA, was sich durch Drücken der Resettaste überprüfen läßt. Das Display zeigt dann die Endadresse EA mit den zugehörigen Daten an.

Datenrücktransport

aus Bit werden Byte

Wir haben gesehen, daß die Bits in Gestalt von zwei hohen Tönen und einem tiefen Ton (logisch Ø) oder einem hohen Ton und zwei tiefen Tönen (logisch 1) auf dem Band stehen. Beim Lesen vom Band müssen die Bit-Zweiklänge in hohe und niedrige Spannungswerte umgesetzt werden, daraus müssen wieder Bits in der vom Computer verstehbaren Form entstehen, diese müssen zu ASCII-Bytes zusammengesetzt und in Daten-Bytes rückgewandelt werden, und schließlich sollen die Daten-Bytes auch noch mit einer bestimmten Adresse beginnend in das RAM gesetzt werden. Das ist zwar eine Menge Arbeit, der Computer schafft sie jedoch mühelos.
Zuerst werden aus den Tonsignalen die Bits rekonstruiert. Beim Lesen vom Band haben die vom Kassettenrekorder kommenden Tonsignale ihre rechteckige Form (siehe ① und ② in Bild 6a sowie Bild 17) verloren, da die Aufnahme-Wiedergabe-Kette einen Tiefpaß bildet. Die am Eingang INPUT des Junior-Computers liegenden Signale sind nun zwar annähernd sinusförmig (siehe Bild 18 und 19), der PLL reagiert jedoch nicht auf die Signalform, sondern auf die Frequenz. Die Frequenzen der Tonsignale, oder anders ausgedrückt, die Lagen der Nulldurchgänge bleiben erhalten. Wie in Kapitel 10 ausführlich beschrieben, erzeugt der PLL aus den Signalen ① und ② in Bild 6a (in der "Sinusversion") die Signale ③ bzw. ④.
Das Ausgangssignal des PLL (③ oder ④) wird invertiert (hoch wird niedrig

und umgekehrt) und dann dem zentralen Teil des Junior-Computers über eine Portleitung zugeführt. Die Kassetten-Lesesoftware vergleicht die Zeitverhältnisse von "High" und "Low" der PLL-Ausgangssignale und entscheidet dann, ob jeweils eine logische Ø oder eine logische 1 auf dem Band steht. Der gleiche Sachverhalt musikalisch ausgedrückt: Ausschlaggebend ist, ob bei den Notentrios die hohen oder die tiefen Noten überwiegen. Innerhalb weiter Grenzen spielt die absolute Tonhöhe ebensowenig eine Rolle wie die absolute Länge der einzelnen Töne. Entscheidend ist nur, ob die Dauer des 3600-Hz-Tons länger als die Dauer des 2400-Hz-Tons ist, oder ob dieses Verhältnis umgekehrt liegt. Aus diesem Grund können Programme und andere Daten, die von einem KIM auf Band gesetzt wurden, vom Junior-Computer problemlos gelesen werden. Die Signale ③ und ④ in Bild 6b haben nämlich die gleiche Form wie die Signale ③ und ④ in Bild 6a; sie unterscheiden sich nur durch die etwa sechsfache zeitliche Dehnung.
Ein Vorteil der sowohl beim Junior als auch beim KIM angewandten Zeitvergleichsmethode ist (sofern der PLL richtig arbeitet) eine weitgehende Unabhängigkeit von Schwankungen und Differenzen der Bandgeschwindigkeit. Differenzen sind praktisch immer vorhanden, wenn für die Aufnahme und die Wiedergabe nicht der gleiche Kassettenrekorder dient, sondern zum Beispiel ein fremdes Programm eingelesen wird. Auch Zeitverhältnisse von 4:3 (statt 2:1) und 3:4 (statt 1:2) verarbeitet die Kassetten-Lesesoftware noch problemlos: An die Stelle von zwei hohen Noten und einer tiefen Note für logisch Ø (siehe Bild 5a) können ohne weiteres vier hohe Noten und drei tiefe Noten treten, ohne daß der Computer die Information mißversteht. Für logisch 1 gilt dies natürlich ebenfalls.
In Kapitel 10 war in Zusammenhang mit der Kassettenhardware vom sogenannten PLL-Jitter die Rede. Es handelt sich dabei um eine Art elektronisches Kontaktprellen: Beim Übergang von "High" nach "Low" (oder umgekehrt) des PLL-Ausgangssignals pendelt dieses kurzzeitig zwischen beiden Zuständen hin und her, bevor der richtige Zustand stabil bleibt. Die Dauer des Jitters wird von der Lese-Software beim Zeitvergleich nicht berücksichtigt, so daß die Verhältnisse von 2:1 für logisch Ø und 1:2 für logisch 1 in Wirklichkeit etwas weniger ausgeprägt sind. Wie schon beschrieben, spielt dies keine Rolle; es kann erst zum Problem werden, wenn man mit wesentlich höheren Schreib- und Lesegeschwindigkeiten arbeiten will. Sobald die Bitzeit in die Größenordnung der Jitterzeit rückt, ist ein zuverlässiger Zeitvergleich nicht mehr möglich.

Suchen, Finden und Lesen

Das Lesen vom Band geschieht mit der Subroutine RDTAPE, die ebenfalls vom Programm TAPE MANAGEMENT aufgerufen wird. Allerdings liest der Junior-Computer nicht einfach das, was nach dem Start des Kassettenrekorders als erstes den Aufnahme-Wiedergabe-Kopf passiert. Zum einen können auf dem Band gesprochene Kommentare aufgenommen sein, zum anderen sind dort meistens mehrere Datenblöcke (z.B. Programme) untergebracht, von denen nur ein bestimmter Datenblock gelesen werden soll. Nach der Eingabe der Programmnummer ID und in einem bestimmten Fall auch einer Startadresse SA muß der gewünschte Datenblock zuerst gesucht

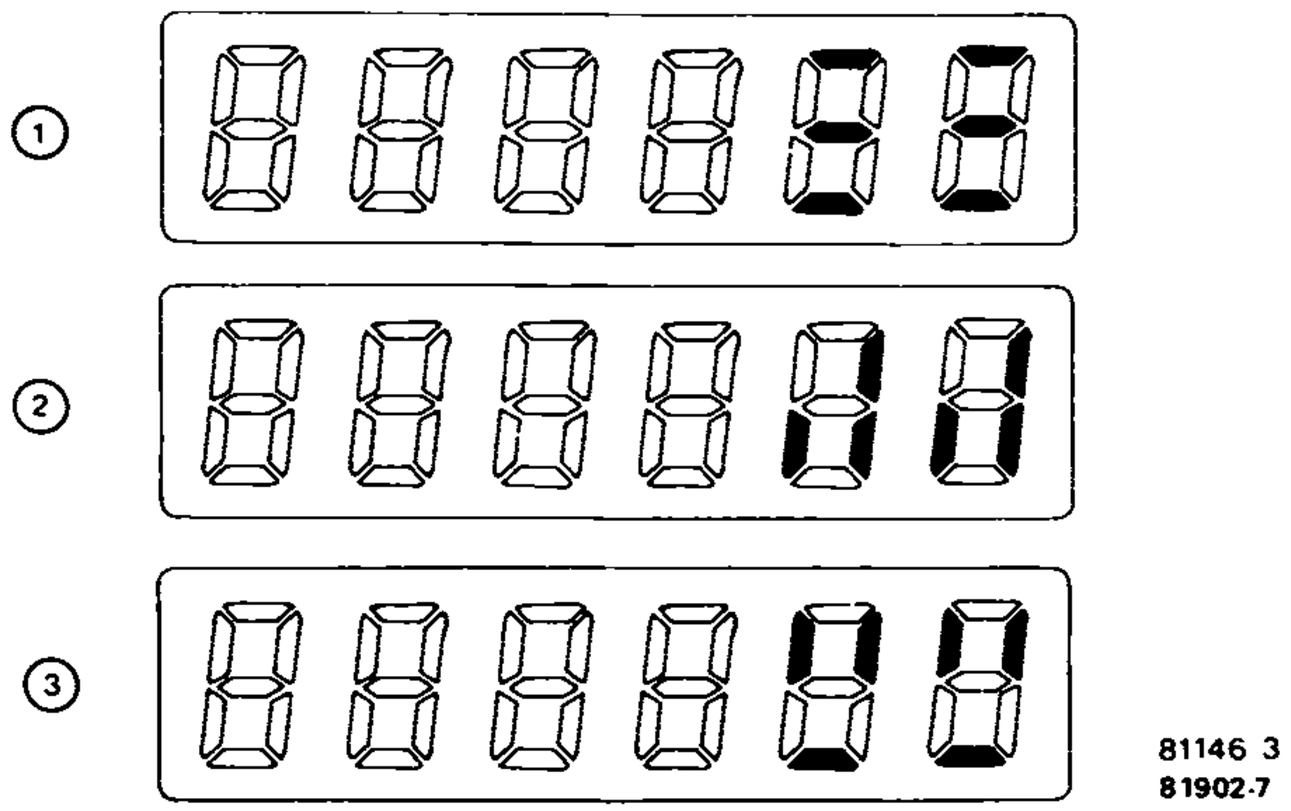


Bild 7. Das Suchen, Finden und Lesen eines Datenblocks vom Band kann auf dem Siebensegmentdisplay verfolgt werden. Was die drei abgebildeten Muster bedeuten, ist im Text erklärt.

und gefunden werden. Über den Stand dieser Dinge geben neben der grünen INPUT-LED, die während des Verbleibs in RDTAPE kontinuierlich aufleuchtet, die beiden rechten Siebensegmentdisplays auf der Basisplatine Auskunft. Bild 7 zeigt die drei möglichen Segmentkonfigurationen; sie haben folgende Bedeutungen:

①

In dieser Weise leuchtet das Display auf, wenn

a. das am Aufnahme-Wiedergabe-Kopf vorbeilaufende Bandstück (Kassettenrekorder in Stellung Wiedergabe) einen Abschnitt zwischen zwei Datenblöcken, gesprochene Kommentare oder überhaupt nichts enthält. Die Anzeige ist dann nicht stabil; das Display flackert;

b. ein Datenblock vom Rekorder wiedergegeben wird, dessen "Einleitung" fehlt, oder, wenn der abgespielte Datenblock nicht die zuvor eingegebene Programmnummer trägt; in diesen Fall ist die Anzeige stabil.

②

Diese Siebensegment-Konfiguration ist immer dann sichtbar, wenn die Synchronisationszeichen gelesen werden, die jedem Datenblock vorangehen. Bei den ersten Zeichen kann das Display flackern, um nach etwa einer Sekunde stabil zu werden. Die 255 vor jedem Datenblock auf dem Band stehenden Synchronisationszeichen passieren den Tonkopf in ungefähr 2,5 Sekunden. Der Junior-Computer stellt sich auf eine ordnungsgemäße Datenübertragung ein, sobald ihn zehn lückenlos aufeinanderfolgende Synchronisationszeichen erreicht haben. Wenn bei den ersten zehn Zeichen eine Störung auftrat, bleiben noch mehr als zwanzig mögliche Anläufe übrig. Der KIM arbeitet dagegen mit nur 100 Synchronisationszeichen, so daß dort die Synchronisation etwas schwieriger ist.

③

Dieses Bild erscheint in der Anzeige, wenn der Datenblock mit der gewünschten Programmnummer gefunden wurde und die Daten in das RAM des Junior-Computers geschrieben werden. Folgt auf ② nicht ③, sondern wieder ① mit stabil aufleuchtenden Displays, so ist der vom Kassetten-

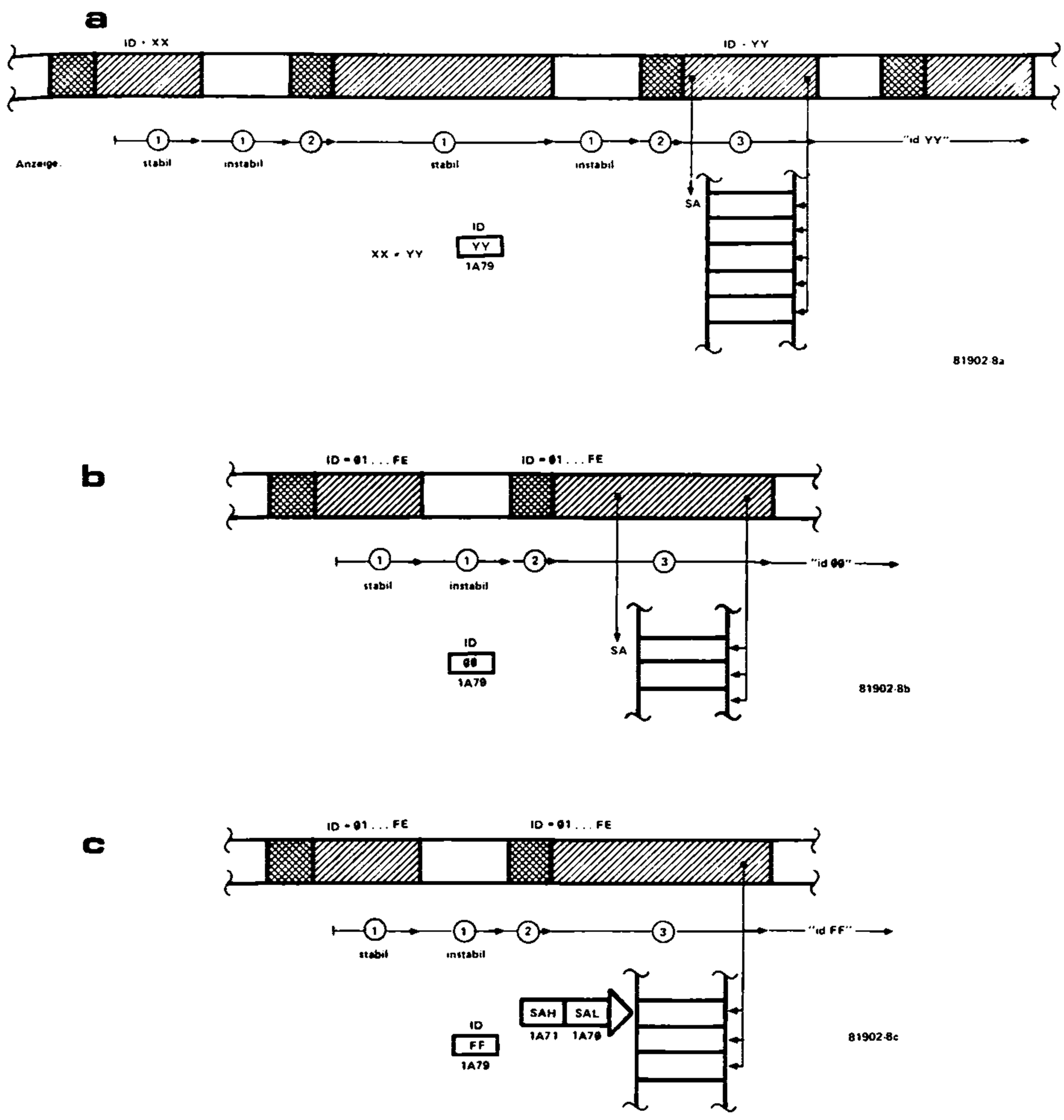


Bild 8. Hier sind schematisch die Vorgänge bei der Suche nach einer Datensendung dargestellt. Der Ablauf hängt davon ab, ob ID = Ø1 . . . FE (Bild 8a), ID = ØØ (8b) oder ID = FF (8c) ist. Nach Drücken von GET startet das Band an einer beliebigen Stelle innerhalb einer Datensendung.

Rekorder wiedergegebene Datenblock nicht der gewünschte. Das Warten auf den nächsten Datenblock läßt sich erheblich abkürzen, wenn außer den Programmnummern auch der jeweilige Stand des Bandzählwerks notiert wurde und der Rekorder im schnellen Vorlauf diese Stelle erreicht.

In Bild 8 ist der Ablauf des Lesevorgangs für die drei Fälle skizziert, daß Programmnummer ID zwischen Ø1 und FE liegt (8a), daß ID = ØØ ist (8b) und daß ID = FF gewählt wurde (8c). Was es mit den beiden zuletzt genannten Programmnummern auf sich hat, das wollen wir nun sehen.

Datenmanipulation

Vor dem Schreiben eines Datenblocks auf Band erhielt dieser ein unverwechselbares Kennzeichen, das ebenfalls auf Band gesetzt wurde: eine Programmnummer, die zwischen Ø1 und FE liegt. Die beiden Zahlen ØØ und FF waren ausgenommen, denn sie haben Sonderfunktionen.

Programmnummer ØØ.

Gibt man vor dem Lesen ØØ als Programmnummer in den Computer ein, dann geschieht folgendes:

- Die nächste vom Kassettenrekorder vollständig wiedergegebene Datensendung wird in das RAM des Juniors gesetzt.
- Das Einlesen in das RAM beginnt bei der Startadresse SA, die auf dem Band steht.
- Die auf dem Band stehende Programmnummer ist ohne Bedeutung.
- Der Ablauf der Vorgänge ist in Bild 8b dargestellt.

Programmnummer FF.

Wählt man FF als Programmnummer, so reagiert der Computer wie folgt:

- Die nächste vom Kassettenrekorder vollständig wiedergegebene Datensendung wird in das RAM des Juniors gesetzt.
- Die RAM-Startadresse, bei der das Einlesen beginnt, ist eine vorher zusammen mit der Programmnummer FF einzugebende, frei wählbare Adresse.
- Die auf dem Band stehende Programmnummer ist ohne Bedeutung.
- Die auf dem Band stehende Startadresse ist ebenfalls ohne Bedeutung.
- Der Ablauf der Vorgänge ist in Bild 8c dargestellt.

Mit FF als Programmnummer ist ein Umsetzen und Aneinanderfügen von beliebigen auf dem Band befindlichen Datenblöcken möglich. Allerdings setzt das die genaue Kenntnis der Bandstellen voraus, an denen die Datenblöcke beginnen (Stand des Bandzählwerks!). Nicht vergessen werden darf, daß beim Datenblocktransfer die absoluten Adreßoperanden teilweise oder ganz geändert werden müssen. Man wird deshalb von Programmnummer FF hauptsächlich dann Gebrauch machen, wenn die betreffenden Datenblöcke (Programme) noch nicht editiert sind. Das heißt, mit ID = FF lassen sich mehrere Files, die mit dem Editor erzeugt wurden, lückenlos aneinanderfügen.

TAPE MANAGEMENT

Umgang mit der Kassetteninterface-Software

Das Programm TAPE MANAGEMENT könnte man auch "Tape Monitor" nennen. Dieses Systemprogramm sorgt für den reibungslosen Ablauf, wenn Daten auf Band geschrieben oder vom Band gelesen werden. Das Programm, das länger als 1024 Byte und kürzer als 2048 Byte ist, befindet sich in einem EPROM vom Typ 2716; dieses EPROM kann daher noch zusätzlich einige "hausgemachte" Bytes aufnehmen. Man kann das EPROM entweder unter Angabe der Nummer ESS 506 vom Elektor-Programmierservice mit TM füllen lassen (näheres siehe aktuelle Ausgabe der Zeitschrift "Elektor") oder das EPROM selbst programmieren; sein Inhalt steht in Form des Hexdump am Schluß dieses Buchs (Anhang 4).

Der Adressenbereich von TAPE MANAGEMENT, nachfolgend kurz TM genannt, umfaßt die Adressen **0800** . . . 0C7F; die Startadresse ist **0810** (nicht **0800**!) In einigen Fällen wird TM über den Editor (siehe Buch 2) verlassen, in anderen Fällen springt man von TM durch Drücken der Taste RST zum Monitor zurück.
Das Systemprogramm TM besteht hauptsächlich aus der Schreib-Subroutine DUMP/DUMPT und der Lese-Subroutine RDTAPE. Der Rest ist Hilfssoftware, die das bequeme Eingeben von **Parametern** über die vorhandenen Tasten AD . . . PC ermöglicht, bevor Daten auf Band geschrieben oder vom Band gelesen werden. In Bild 9 ist das Keyboard mit den bisherigen und den neu hinzugekommenen Funktionen der Steuertasten gezeichnet. Wir wollen nun die neuen Funktionen der Tasten +, AD, PC, DA und GO der Reihe nach besprechen.

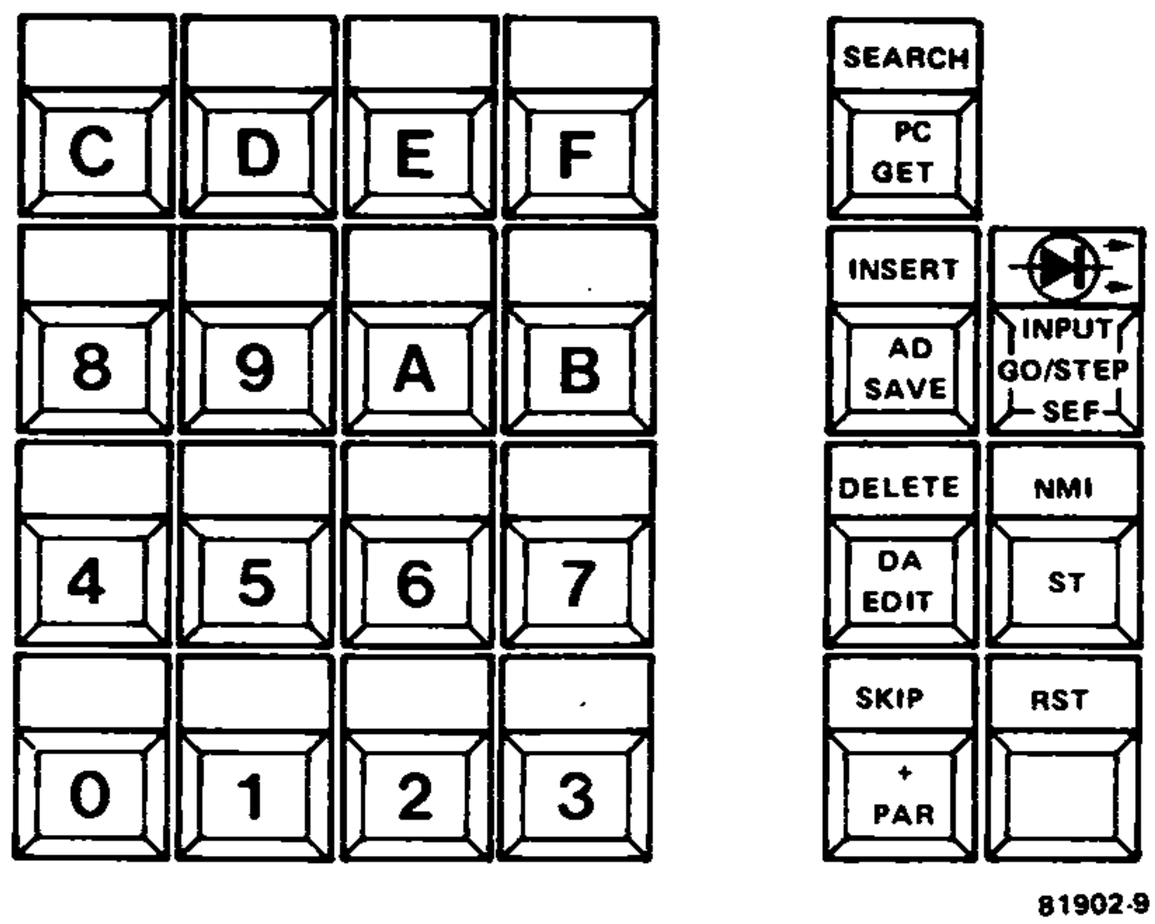


Bild 9. Das hexadezimale Keyboard des Junior-Computers. Alle Steuertasten (außer RST und NMI) erhalten durch TM eine dritte Funktion.

1 Taste PAR (sonst + bzw. SKIP)

Parametereingabe

Beginnen wir wie gewohnt mit einem Beispiel für die Bedienung des Keyboards:

Eingabe	Anzeige	
RST 0 8 1 0 GO	id 00	(① in Bild 10)
2 C	id 2C	
PAR	SAH 00	(② in Bild 10)
0 2	SAH 02	
PAR	SAL 00	(③ in Bild 10)
PAR	EAH 00	(④ in Bild 10)
0 3	EAH 03	
PAR	EAL 00	(⑤ in Bild 10)
F F	EAL FF	

```
PAR          bEGHXX (⑥ in Bild 10)
PAR          bEGLXX (⑦ in Bild 10)
PAR          EndHXX (⑧ in Bild 10)
PAR          EndLXX (⑨ in Bild 10)
PAR          id    2C
PAR          SAH Ø2
PAR          SAL ØØ
PAR          EAH Ø3
Ø 4          EAH Ø4
PAR          EAL FF
Ø Ø          EAL ØØ
PAR          bEGHXX
```

und so weiter.

Hier wurde das Schreiben einer Datensendung auf Band vorbereitet, denn wir haben in den Computer eine Programmnummer ID (angezeigt wird "id" statt "ID"), eine Startadresse SA und eine Endadresse EA eingegeben. Auf das Band gesetzt wird der Datenblock, der im RAM von Adresse Ø2ØØ bis einschließlich Adresse Ø3FF steht; als Programmnummer haben wir 2C gewählt. Noch rechtzeitig wurde ein Fehler bei der Eingabe korrigiert: Da der Inhalt des RAM-Bereichs Ø2ØØ . . . Ø3FF auf Band kopiert werden soll, lautet die Endadresse EA nicht Ø3FF, sondern Ø4ØØ (EA = LA + 1)! Durch wiederholtes Drücken von Taste PAR wurden EAH und EAL ein zweites Mal auf das Display gebracht, so daß der Fehler beseitigt werden konnte.
An dem vorstehenden Beispiel für die Eingabe der Parameter, die der Computer vor Schreiben einer Datensendung auf Band benötigt, wird folgendes deutlich:

- Alle neun Parameter, die für die vier übrigen Tastenfunktionen vorbereitenden Charakter haben, sind durch wiederholtes Drücken von Taste PAR zugänglich; sie lassen sich spezifizieren, indem man nacheinander zwei numerische Tasten drückt.
- Nach dem Start von TM sind die Parameter ID, SAH, SAL, EAH und EAL sämtlich Null. Die restlichen (Editor-)Parameter haben einen willkürlichen Wert, wenn der Computer unmittelbar vor dem TM-Start eingeschaltet wurde, oder rühren vom vorangegangenen Geschehen her (Verbleib im Editor). In Bild 10 wurden bei den Anzeigedarstellungen ⑥ . . . ⑨ die Daten "88" zum Zeichen dafür gezeichnet, daß beliebige Segmente aufleuchten können (Don't Care).
- Die Parameter erscheinen im Rollmodus auf der Anzeige: Bei fortlaufendem Drücken der PAR-Taste folgt auf den letzten Parameter wieder der erste.
- Sobald man eine numerische Taste drückt, wird der angezeigte Parameter überschrieben. Die hexadezimalen Ziffern schieben sich wie beim Datenmodus des Monitors von rechts nach links in die beiden Datendisplays ein.
- Wenn der Parameter eine Adresse ist, erscheint zuerst das höhere Adreßbyte (ADH) und dann das niedrige Adreßbyte (ADL) auf den Displays.

In dieser Reihenfolge ist die Eingabe besonders einfach, da sie der gewohnten Schreibweise entspricht.
Die Tastenfunktion PAR dient eigentlich nur der Bequemlichkeit, denn auch mit den Tastenfunktionen AD, DA und + des Monitors ist die Ein-

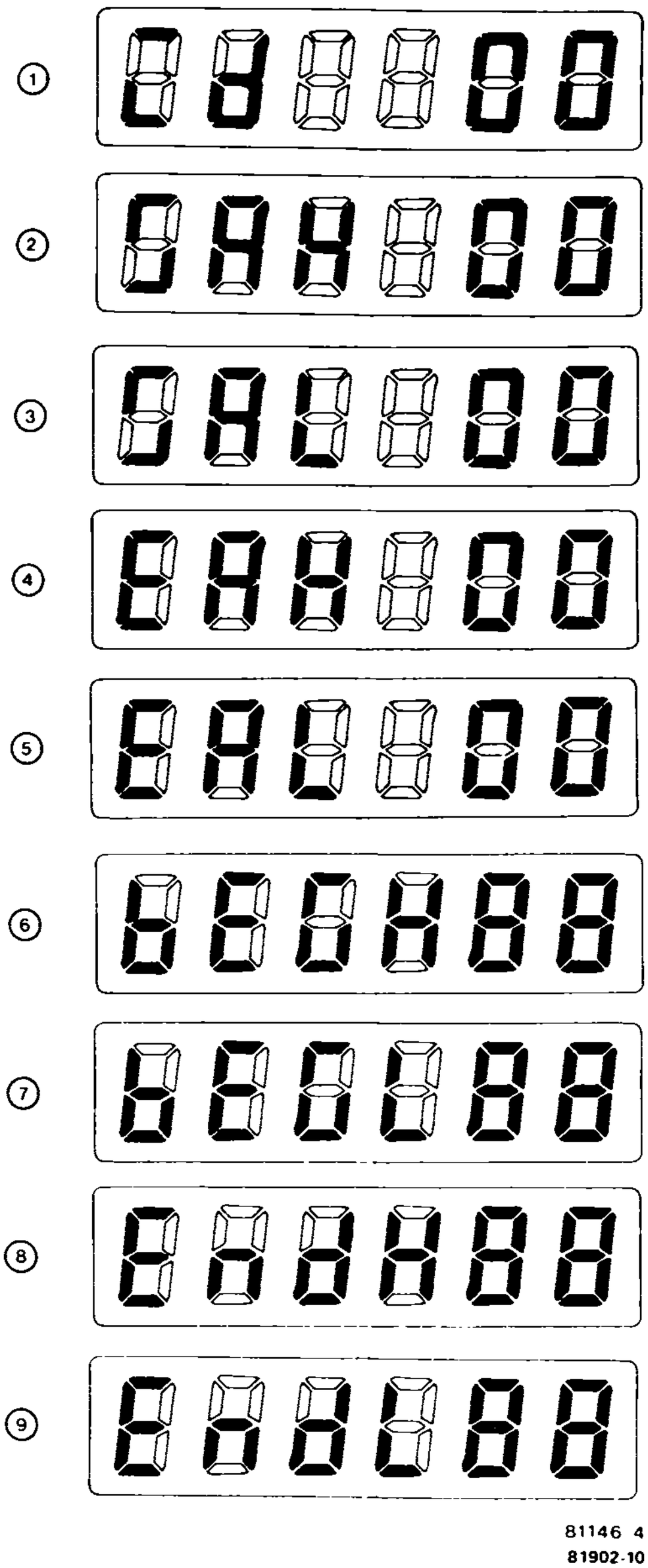


Bild 10. Die neun Parameter, die den Datentransport zum und vom Band betreffen, werden von TM einzeln sichtbar gemacht. Wenn Taste PAR gedrückt wird, wechselt die Anzeige von einem Parameter und seinen Daten zum nächsten über.

Tabelle 1. Liste der Speicherplätze, die TM in Seite 1A belegt. Die Variablen dürfen ebenso wie die vom Monitor, Editor und Assembler niemals während des Lesens einer Datensendung überschrieben werden.

TEMPORARY DATA BUFFERS			
SY	*	$1A69	SYN COUNTER
BYTE	*	$1A6A	BYTE FROM TAPE
CHAR	*	$1A6B	CHARACTER FROM TAPE
HIGHER	*	$1A6C	3600 Hz HALF PERIODE DELAY
LOWER	*	$1A6D	2400 Hz HALF PERIODE DELAY
CHKL	*	$1A6E	CHECK SUM LOW
CHKH	*	$1A6F	CHECK SUM HIGH
SAL	*	$1A70	START ADDRESS LOW
SAH	*	$1A71	START ADDRESS HIGH
EAL	*	$1A72	END ADDRESS LOW
EAH	*	$1A73	END ADDRESS HIGH
SYNCNT	*	$1A74	
BITS	*	$1A75	AMOUNT OF BITS
FIRST	*	$1A76	HALF PERIODE AMOUNT OF 3600 Hz
SECOND	*	$1A77	HALF PERIODE AMOUNT OF 2400 Hz
GANG	*	$1A78	TEMP OF PBD-BITS
ID	*	$1A79	ID OF THE DUMPT PROGRAM
NMI	*	$1A7A	NMI VECTOR

gabe der zu den Lese- und Schreibsubroutinen von TM gehörenden Parametern möglich. Alle neun Parameter stehen nämlich, wie aus Tabelle 1 hervorgeht, in Speicherplätzen auf den Seiten 00 oder 1A. Trotzdem ist die Tastenfunktion PAR keineswegs überflüssig, denn ohne PAR wären wahrscheinlich Verwechslungen nicht selten: Steht in Adresse 00E2 nun BEGADL oder BEGADH? Mit Tastenfunktion PAR erübrigen sich solche Überlegungen.

2 Taste SAVE (sonst AD bzw. INSERT)

Datenkonservierung

Das Wort "save" bedeutet so viel wie "retten" oder "bewahren". Mit dieser Tastenfunktion wird ein Datenblock, verpackt als Datensendung, auf folgende Weise zum Kassettenrekorder geschickt und dort auf Band gesetzt:

- Eingabe einer Programmnummer ID (01 . . . FE, nicht 00 oder FF!);
- Eingabe der Startadresse SA;
- Eingabe der Endadresse EA, die um eine höher als die letzte Adresse (LA) des Datenblocks liegt;
- In den Kassettenrekorder wird eine leere oder zu löschende Kassette eingelegt;
- Notiert werden folgende Angaben: der Programm- oder Datenblocktitel, die Programmnummer ID, die Start- und Endadresse SA, EA sowie der Stand des Bandzählwerks. Diese Angaben können auch vor Beginn der Datensendung auf Band gesprochen werden.

- Start des Kassettenrekorders in der Betriebsart "Aufnahme". Zu achten ist auf richtige Aussteuerung, sofern der Kassettenrekorder nicht mit einer Aussteuerungsautomatik ausgerüstet ist.
- Nach dem eventuellen Aufsprechen des Titels usw. wird Taste SAVE gedrückt. Der Computer setzt daraufhin mit Subroutine DUMP/DUMPT den spezifizierten Datenblock auf Band. Solange dies geschieht, leuchtet die rote OUTPUT-LED (D5) auf; die Siebensegmentdisplays verlöschen. Nach Ende der Datensendung meldet sich der Junior-Computer mit "id XX" zurück, wobei XX die gewählte Programmnummer ist.
- Stop des Kassettenrekorders.

So wird in der Praxis ein Datenblock vom RAM auf eine Kassette kopiert. Wenn der Antriebsmotor des Kassettenrekorders über dessen Fernbedienungsanschluß vom Computer gesteuert wird, startet der Rekorder beim Drücken der Taste SAVE automatisch; vorher ist der Rekorder selbstverständlich auf "Aufnahme" zu schalten! Zum Aufsprechen des Kommentars muß der Motor natürlich ebenfalls laufen. Ist die Datensendung des Computers beendet, kann Resettaste RST gedrückt werden. Von den Displays wird nun Endadresse EA mit den zugehörigen Daten angezeigt. Dies ist gleichzeitig die Startadresse SA eines anschließenden, aber eventuell erst später zu übertragenden Datenblocks.

3 **Taste GET** (sonst PC bzw. SEARCH)

Gezielter Zugriff

Durch Drücken dieser Taste geben wir dem Junior-Computer den Befehl, mit Hilfe der Subroutine RDTAPE einen bestimmten Datenblock vom Band zu lesen und in das RAM des Computers zu setzen. Wir bedienen den Computer wie folgt:

- Eingabe der Programmnummer ID. Die möglichen Zahlenwerte reichen von ØØ bis FF, obwohl auf dem Band nur Programmnummern zwischen Ø1 und FE vorkommen. Gibt man keine Programmnummer ein, dann bleibt der bisherige Inhalt von Speicherplatz ID gültig (① in Bild 10).
- Wenn ID = FF gewählt wird, muß auch eine Startadresse SA spezifiziert werden. Fehlt diese Eingabe, dann ist SA gleich dem bisherigen Inhalt von SAH und SAL (② und ③ in Bild 10). Nach dem Start sind SAH und SAL Null, so daß die Startadresse ØØØØ ist.
- In den Kassettenrekorder wird die Kassette eingelegt, auf der sich der einzulesende Datenblock befindet. Wurde eine Programmnummer zwischen Ø1 und FE gewählt, muß die Bandstelle nicht unbedingt bekannt sein. Der Computer liest den gewünschten Datenblock ein, sobald dieser den Tonkopf passiert. Im Fall ID = ØØ oder ID = FF ist dagegen der Rekorder genau am Beginn des Datenblocks zu starten, denn der Computer liest dann den nächstfolgenden Datenblock ohne Berücksichtigung der Programmnummer ein.
- Nun wird Taste GET gedrückt, so daß die grüne INPUT-LED aufleuchtet.
- Start des Kassettenrekorders in Stellung Wiedergabe. Auf den Displays erscheinen die in Bild 7 dargestellten Segmentkonfigurationen. Wenn die Displays in der mit ③ bezeichneten Weise aufleuchten, findet der Transport des Datenblocks vom Band in das RAM statt. Nachdem dies

beendet ist, meldet sich der Junior-Computer mit "id XX" zurück, wobei XX die vorher eingegebene Programmnummer ist. Hat der Computer bei der Kontrolle durch Vergleich der CHKL- und CHKH-Werte einen Übertragungsfehler festgestellt, bleibt er in der Subroutine RDTAPE; in diesem Fall erscheint keine Rückmeldung "id XX" auf den Displays.
Auch nach Einlesen eines Datenblocks kann man die einzelnen Parameter auf den Displays erscheinen lassen, wenn man Taste PAR drückt. Von den angezeigten Daten bezieht sich jedoch nur die Programmnummer ID (①) und im Fall ID = FF auch die Startadresse SAH/SAL (② und ③) auf den gerade eingelesenen Datenblock. Die Endadresse EAH/EAL (④ und ⑤ in Bild 10) stimmt normalerweise nicht mit der Endadresse dieses Datenblocks überein; sie ist nur beim Schreiben auf das Band (SAVE) von Bedeutung.
Drückt man unmittelbar nach dem Einlesen eines Datenblocks die RST-Taste (Rücksprung zum Monitor), dann wird die zu diesem Datenblock gehörende Endadresse EA zusammen mit den dort stehenden Daten sichtbar (EA ergibt sich indirekt aus der Startadresse SA und der Länge des Datenblocks). Dies gilt, solange nicht ID = FF war. Wurde FF als Programmnummer gewählt, so geschieht nach dem Einlesen zusätzlich noch folgendes: Der Inhalt des Adreßpointers POINT, mit dessen Hilfe der Datenblock in den Speicher gelangt ist, wird um 1 vermindert; anschließend wird die Startadresse SAH, SAL mit POINT gleichgesetzt. Drückt man jetzt die RST-Taste, dann erscheint die letzte Adresse LA des gerade eingelesenen Datenblocks mit den zugehörigen Daten auf den Displays.
Weshalb wird POINT um 1 vermindert und die Startadresse SA gleich POINT gesetzt? Antwort: Um aus mehreren Datensendungen bestehende **editierte Programme** (die noch nicht assembliert sind und deshalb noch Labels enthalten) nahtlos aneinanderfügen zu können. **Will man gewöhnliche Programme aneinanderfügen, so muß die neue Startadresse um 1 erhöht werden. Anderenfalls wird nämlich das letzte zum vorangehenden Datenblock gehörende Byte vom ersten Byte des folgenden Datenblocks überschrieben!**
In Bild 11 ist dies anschaulich dargestellt. Mit einer Startadresse SA, die gleich BEGAD ist, und mit einer Endadresse EA, die mit CEND übereinstimmt, wird ein editiertes Programm als Datensendung auf Band gesetzt (siehe auch Beschreibung der Tastenfunktion SEF). Der variable Endadressenpointer CEND ist auf den nächstfolgenden freien Speicherplatz gerichtet, während einen Platz höher das EOF-Zeichen 77 steht. Sollen mehrere editierte Blöcke aneinandergefügt werden, dann müssen die dazwischenliegenden EOF-Zeichen verschwinden, ohne daß Löcher entstehen. Dies ist die Erklärung für das Überlappen um einen Speicherplatz, das durch Herabsetzen von POINT um 1 erreicht wird, sowie für das anschließende Gleichsetzen der neuen Startadresse mit POINT.

4 **Taste EDIT** (sonst DA bzw. DELETE)

Editor-Kaltstart

Diese Tastenfunktion ist mit der Eingabe AD 1 C B 5 GO identisch und hat den **Kalt**start des Editors zur Folge. Die Beginn- und Endadressen

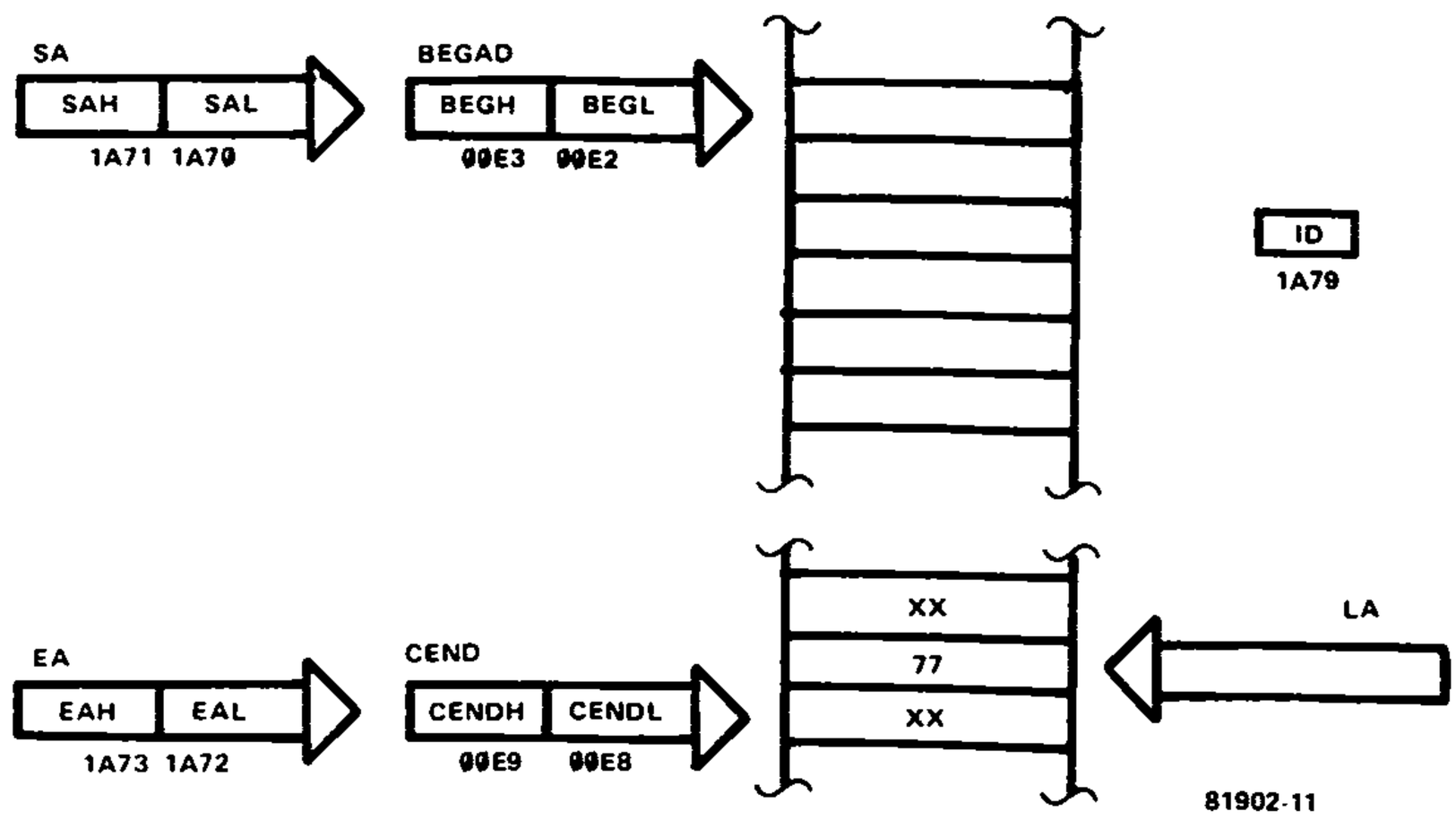


Bild 11. Gleiche Darstellung wie in Bild 2; hier wird jedoch durch Drücken von SEF ein editierter Datenblock von TM auf Band gesetzt und nach Drücken von GET wieder zurücktransportiert.

BEGAD und ENDAD können dem Computer nun bequem über die Tastenfunktion PAR mitgeteilt werden (⑥ . . . ⑨ in Bild 10). Nach Drücken von EDIT erscheint das EOF-Zeichen 77 auf den Displays. Im Prinzip ist dies nichts Neues, sondern EDIT erleichtert lediglich die für den Editor-Kaltstart notwendige Bedienung.

5 Taste SEF (sonst GO bzw. INPUT)

Rettung des Originals

SEF und SAVE haben sehr ähnliche Funktionen. SEF bedeutet "Save Edited File" oder "schreibe ein editiertes Programm auf Band": das sind sämtliche Daten ab Adresse BEGAD (= SA) bis Adresse CEND (= EA) einschließlich des EOF-Zeichens 77 (siehe Bild 11). Deshalb erscheint die Zahl 77 auf den Displays, wenn man nach Schreiben des Datenblocks auf Band die RST-Taste drückt.

Die Tastenfunktion SEF setzt voraus, daß unmittelbar vorher mit dem Editor gearbeitet wurde und daß der variable Endadreßpointer CEND in Übereinstimmung mit dem aktuellen Stand ist. Der Junior-Computer darf nicht zwischenzeitlich ausgeschaltet worden sein.

Um ein editiertes Programm auf Band zu setzen, gehen wir wie folgt vor:

- Sprung mit Hilfe des Monitors nach TM: RST 0 8 1 0 GO XX

Nach Drücken der GO-Taste startet TM, so daß Muster ① aus Bild 10 auf den Displays erscheint. XX steht für die gewählte Programmnummer; sie ist als einziges einzugeben, da die Startadresse SA und die Endadresse EA über BEGAD bzw. CEND festliegt.

- In den Kassettenrekorder wird die Kassette eingelegt, auf die das Programm aufgenommen werden soll. Gegebenenfalls läßt man das Band bis zum Ende der dort schon stehenden Aufzeichnungen vorlaufen.

- Der Stand des Bandzählwerks und die gewählte Programmnummer werden sofort, die Start- und Endadressen SA und EA jedoch **erst nach beendeter Datensendung** notiert. Letzteres hat seinen Grund darin, daß SAH, SAL, EAH und EAL erst nach Drücken von SEF mit BEGADH, BEGADL, CENDH und CENDL geladen werden.
- Der Rekorder wird in Stellung "Aufnahme" gestartet. Auf richtige Aussteuerung achten!
- Nach Drücken von SEF setzt der Junior-Computer mit Subroutine DUMP/DUMPT das editierte Programm auf Band. Währenddessen leuchtet die rote OUTPUT-LED (D5) auf; die Siebensegmentdisplays verlöschen. Nach getaner Arbeit meldet sich der Computer mit der ersten Instruktion des auf Band gesetzten editierten Programms zurück, weil nach der Datensendung der Displaypointer CURAD gleich BEGAD gesetzt wird und dann ein Sprung zum Editor folgt **(Warmstart).**
- Stop des Kassettenrekorders.
- Jetzt werden SA und EA des gerade übertragenen Datenblocks notiert. Es gilt: SAH = BEG(AD)H, SAL = BEG(AD)L, EAH = END(AD)H und EAL = END(AD)L. **Diese Werte werden nur benötigt, wenn nach der Rückübertragung des Editor-Datenblocks vom Band noch vom Editor (Warmstart) Gebrauch gemacht werden muß, nicht aber, wenn der Datenblock sofort nach dem Lesen vom Band assembliert wird.** Die Daten von SA und EA erhält man wie folgt:

RST	Rücksprung vom Editor zum Monitor
Ø 8 1 Ø GO	(ID)
PAR	(SAH)
PAR	(SAL)
PAR	(EAH)
PAR	(EAL)
PAR PAR	(BEGH) (BEGL); siehe SAH und SAL
PAR PAR	(ENDH) (ENDL); eventuell zusätzlich notieren

Will man nach Notieren dieser Daten zum Editor **mit einem Warmstart** zurückspringen (dort wäre man nach SEF ohne den zum Notieren notwendigen Sprung in den Monitor), dann genügt folgende Eingabe:
RST 1 C C A GO.
BEGAD und ENDAD brauchen nicht eingegeben werden. Der variable Adreßpointer CEND zeigt auf den leeren Speicherplatz unterhalb des Platzes mit dem EOF-Zeichen 77, während der Displaypointer CURAD mit BEGAD zusammenfällt: Auf den Displays ist die erste Instruktion des Editor-Datenblocks zu sehen. Voraussetzung ist natürlich, daß der Junior-Computer nicht zwischendurch ausgeschaltet wurde!
Aus welchem Grund ist es in bestimmten Fällen zweckmäßig, die Startadresse SA und die Endadresse EA zu notieren? Das soll nachfolgend beleuchtet werden.

Eine wichtige SEF-Anwendung

Gesetzt den Fall, wir haben ein Programm entwickelt und setzen dieses Programm nun mit Hilfe des Editors in das RAM des Junior-Computers. Zu gegebener Zeit ist der Moment gekommen, daß wir von der Funktionsfähigkeit des Programms überzeugt sind. Wir lassen das Programm daher

vom Computer assemblieren, um es anschließend zu testen. Leider stellt sich dabei heraus, daß uns doch ein Fehler unterlaufen ist (welches Programm läuft schon auf Anhieb!). Nun sind aber die Label beim Assemblieren aus dem Programm entfernt worden, so daß es sich für die Nachbearbeitung mit dem Editor nicht mehr eignet. Es bleibt nichts anderes übrig als das Programm von Anfang an neu zu editieren (hoffentlich wurde alles dokumentarisch auf Papier festgehalten!), oder aber wir haben in kluger Voraussicht **das editierte, aber noch nicht assemblierte Programm auf Band gespeichert.** Dann bleibt es uns erspart, das Programm noch einmal editieren zu müssen.

Und so wird das noch nicht assemblierte Programm für die eventuelle Nachbearbeitung "eingefroren":

- Vor dem Assemblieren und anschließenden Testen wird das Programm wie im Abschnitt "Tastenfunktion SEF" beschrieben auf Band gesetzt;

Programmnummer ID, Startadresse SA und Endadresse EA werden notiert.

- Nach der Konservierung auf Band mit SEF und Notieren der Parameter verläßt man TM und springt über den Monitor zum Assembler:

RST 1 F 5 1 GO.

Das Programm wird gestartet: AD (Startadresse) GO. Bekanntlich kann nun zweierlei passieren. Entweder der Computer tut was er soll, oder aber das Programm ist fehlerhaft, so daß der Computer etwas tut, was er nicht soll.

- Wenn das Programm nicht wie gewünscht läuft, muß zuerst der Fehler aufgespürt werden. Das kann naturgemäß einige Zeit in Anspruch nehmen. Wir schalten deshalb den Junior-Computer aus, denn auch das spart Energie. Der Verlust seines "Gedächtnisses" ist nicht weiter schlimm, denn wir haben alles Notwendige notiert.
- Wir haben den Fehler gefunden und wollen das Programm in seiner editierten Form korrigieren. Dazu schalten wir den Junior-Computer wieder ein, drücken RST Ø 8 1 Ø GO X Y (XY ist die Programmnummer ID) und verfahren ferner so, wie im Abschnitt "Tastenfunktion GET" beschrieben. Danach wird das Keyboard wie folgt bedient:

```
RST ØØE 2
DA PP          PP = BEGADL  = notiertes SAL-Byte
+  QQ          QQ = BEGADH  = notiertes SAH-Byte
+  RR          RR = ENDADL  = notiertes ENDADL-Byte
+  SS          SS = ENDADH  = notiertes ENDADH-Byte
+  PP          CURADL = BEGADL
+  QQ          CURADH = BEGADH
+  TT          CENDL  = notiertes EAL-Byte
+  UU          CENDH  = notiertes EAH-Byte
AD 1 C C A GO  Warmstart des Editors
```

Da CURAD gleich BEGAD gesetzt wurde, erscheint nach dem Warmstart des Editors die erste Instruktion (oder das erste Label) des vom Band zurücktransportierten, editierten Programms auf den Displays. Mit dem Editor können jetzt die Korrekturen vorgenommen werden. Danach wird das korrigierte, noch nicht assemblierte Programm wieder auf Band gesetzt (es könnte noch ein Fehler unentdeckt geblieben sein), und notfalls wird das Ganze noch einmal wiederholt.

- Läuft das Programm einwandfrei, dann kann die editierte Version vom

Band gelöscht werden, es sei denn, daß das Programm später noch erweitert werden soll. Das fehlerfreie und inzwischen assemblierte Programm wird in jedem Fall auf Kassette konserviert. Ferner ist eine schriftliche Programmdokumentation bei späteren Aktivitäten äußerst hilfreich. Man kann ein Hexdump anfertigen oder ausdrucken lassen (siehe Kapitel 12) oder das Programm in assemblierter Form dokumentieren. Im dabei entstehenden Listing sind die Instruktionen tabellarisch der Reihe nach aufgeführt, wobei eine Spalte für die Label, eine Spalte für die Bytes, eine Spalte für die Mnemonics und eine Spalte für Kommentare (Erläuterungen) reserviert ist. Beim Aufstellen des Listings leistet der Editor wertvolle Hilfe, denn mit ihm kann das Programm instruktionsweise durchlaufen werden (Drücken der SKIP-Taste). Dies wurde in Kapitel 5 (Buch 2) ausführlich behandelt; hier soll deshalb nur eine kurze Gebrauchsanweisung gegeben werden;

- CURAD wird gleich BEGAD gesetzt, dies ist die Startadresse des Programms.
- CEND wird gleich ENDAD gesetzt. ENDAD kann sich durch die Fehlerkorrektur verschoben haben. Deshalb ist zu prüfen, ob

CEND = ENDAD auf eine Adresse gerichtet ist, die um mindestens zwei Schritte höher liegt als die letzte Instruktion des Programms. Anderenfalls kann das Programm mit der SKIP-Taste nicht vollständig durchlaufen werden; es folgt dann die Fehlermeldung EEEEEE.

- Der Editor wird **warm** gestartet. Nun läßt sich mit Taste SKIP eine Instruktion nach der anderen auf den Displays sichtbar machen.

Achtung! In allen Fällen, in denen mit einem vorhandenen editierten Datenblock gearbeitet wird, darf der Editor **niemals kalt** gestartet werden, zum Beispiel durch Drücken von Taste EDIT! Ein Kaltstart ist nur dann notwendig, wenn mit dem Editieren von Grund auf neu begonnen wird. Beim Kaltstart eines bereits editierten Blocks werden ein oder mehrere Speicherplätze vom EOF-Zeichen 77 überschrieben, während das EOF-Zeichen beim Warmstart ganz am Schluß steht und keine neue 77 vom Anfang her durchgeschoben wird.
Noch eine Anmerkung: Wenn mit ID = FF mehrere editierte Datenblöcke eingelesen werden, an deren Ende jeweils das EOF-Zeichen 77 steht, gilt CEND als Endadresse des zuletzt eingelesenen Datenblocks! Man denke auch an die Anpassung von ENDAD. Mit Programmnummer FF lassen sich die Datenblöcke beim Rücktransport vom Band beliebig verschieben. Selbstverständlich müssen dann auch die zuständigen Parameter geändert werden.

PLL-Einstellung

Fehlerfreies Lesen

Ebenso wie der Mensch mit seinen Augen liest, so liest der Junior-Computer mit Hilfe des zum Kassetteninterface gehörenden PLL-ICs. Deshalb ist die PLL-Schaltung mit Sorgfalt so einzustellen, daß der Junior weder an Kurz- noch an Weitsichtigkeit leidet: Jedes Bit, das auf Band steht, muß seinen ursprünglichen Wert Ø oder 1 behalten.
Der PLL wird mit dem Mehrgangpoti P1 eingestellt, das sich auf der Inter-

facekarte befindet (siehe Bild 2 von Kapitel 10). Dieses Poti bestimmt die Frequenz des PLL-internen Oszillators (VCO) bei fehlendem PLL-Eingangssignal. Solange P1 nicht allzu weit von seiner Mittelstellung entfernt steht, liegt die VCO-Leerlauffrequenz irgendwo zwischen 2400 Hz und 3600 Hz. Die VCO-Frequenz folgt dann der Frequenz des vom Kassettenrekorder kommenden Signals; das PLL-Ausgangssignal ist abwechselnd "high" und "low". Fällt die VCO-Leerlauffrequenz außerhalb dieses Bereichs, arbeitet das PLL-System nicht; es ist dann nicht "eingerastet". In diesem Fall bleibt das Ausgangssignal kontinuierlich hoch oder niedrig. **P1 muß deshalb vor der Einstellung ungefähr in Mittelstellung gebracht werden.**
Die PLL-Einstellung wird so vorgenommen, daß das Zeitverhältnis von "high" zu "low" des PLL-Ausgangssignals möglichst exakt 2:1 (logisch Ø) bzw. 1:2 (logisch 1) ist. Zwei Wege führen zu diesem Ziel:

- Von einem Band werden Testdaten gelesen, wobei die korrekte oder fehlerhafte Arbeitsweise auf den Displays sichtbar gemacht wird. Poti P1 ist in die Mitte des Einstellbereichs zu setzen, in dem die Testdaten fehlerfrei gelesen werden.
- Von einem Band werden Testdaten gelesen, das PLL-Ausgangssignal wird jedoch von einem Oszilloskop sichtbar gemacht. An Hand des Schirmbildes werden mit P1 die geforderten Zeitverhältnisse eingestellt.

Noch eine Randbemerkung: In einem der vorangegangenen Abschnitte dieses Kapitels wurde auf die hohe Störsicherheit hingewiesen, die die Zeitvergleichsmethode bei der Speicherung von Daten auf Band bietet. Das bedeutet jedoch nicht, daß P1 ungefähr in Mittelstellung stehenbleiben kann und eine genauere Einstellung überflüssig ist. Nachfolgend werden deshalb die einzelnen Schritte in aller Ausführlichkeit beschrieben.

1. PLL-Einstellung ohne Oszilloskop

Voraussetzung für die PLL-Einstellung, gleichgültig ob mit oder ohne Oszilloskop, ist natürlich ein funktionsfähiger Junior-Computer einschließlich der Interfacekarte; außerdem muß IC4 mit TM geladen sein. Ohne TM ist die PLL-Einstellung nutzlos, es sei denn, man hat eigene Kassettensoftware entwickelt und diese in IC4 geladen. Wir gehen davon aus, daß TM arbeitet: Nach Eingeben von RST Ø 8 1 Ø GO erscheint "id ØØ" auf den Displays. Nun können die Vorbereitungen für die PLL-Einstellung beginnen:

1. Wir geben die beiden Hilfsprogramme aus Tabelle 2 in den Junior-Computer ein. Das Programm von Adresse Ø2ØØ bis Ø25Ø verwendet eine Subroutine von TM und setzt damit vier Minuten lang das Synchronisationszeichen (hexadezimal 16) auf Band. Danach folgt ein Rücksprung zum Monitor. Das im Speicherbereich Ø251 . . . Ø283 stehende Programm benutzt vier Subroutinen von TM; es dient zum Lesen der Synchronisationszeichen vom Band und zur visuellen Information über das Leseergebnis. Auf den Displays erscheint entweder Muster ① oder ② der in Bild 7 dargestellten Segmentkonfigurationen.
2. In den Rekorder wird eine Kassette eingelegt und dieser in Stellung "Aufnahme" gestartet. Trimmpoti P2 auf der Interfacekarte steht in Mittelstellung; die Aussteuerung des Kassettenrekorders, sofern von außen beeinflußbar, ebenfalls.

Tabelle 2. Zwei Hilfsprogramme zum Schreiben bzw. Lesen der Testdaten, die für das Einstellen des PLL an Hand der Displayanzeige benötigt werden.

```
M
HEXDUMP: 200,250
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0200: A9 7D 8D 6C 1A A9 C3 8D 6D 1A A9 03 8D 76 1A A9
0210: 02 8D 77 1A A9 47 A2 FF 8D 82 1A 8D 78 1A 8E 83
0220: 1A A9 00 A2 7F 8D 80 1A 8E 81 1A A9 DD 8D 00 1A
0230: 8D 01 1A 18 A9 01 6D 00 1A 8D 00 1A A9 00 6D 01
0240: 1A 8D 01 1A B0 08 A9 16 20 A3 0A 4C 33 02 4C 1D
0250: 1C
JUNIOR
M
HEXDUMP: 251,283
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0251: A9 32 8D 82 1A 8D 78 1A A9 7E 8D 83 1A A9 7F 8D
0261: 81 1A A9 FF 8D 6B 1A 20 C2 0B 6E 6B 1A AD 6B 1A
0271: 20 E8 0B C9 16 D0 F0 20 36 0C 20 5D 0C C9 16 F0
0281: F6 D0 DF
JUNIOR
```

3. Es folgt die Eingabe AD 0 2 0 0 GO über das Keyboard. Daraufhin werden die Synchronisationszeichen auf das Band geschrieben; gleichzeitig leuchtet die rote OUTPUT-LED (D5) auf.
4. Nach ca. vier Minuten ist der Schreibvorgang beendet. Die rote LED verlischt, und auf den Displays erscheint "0200A9". Das Band wird gestoppt und bis zum Beginn der Synchronisationszeichen zurückgespult.
5. Nun wird der Kassettenrekorder in Stellung "Wiedergabe" gestartet (beim Anschluß des Kassetteninterface an den Lautsprecherausgang mittlere Lautstärke einstellen), und über das Keyboard wird AD 0 2 5 1 GO eingegeben. Wenn wir das Band zu weit zurückgespult haben, ist auf den Displays mit Sicherheit Muster ① aus Bild 7 zu sehen, vorausgesetzt P1 steht ungefähr in Mittelstellung. Sobald die Synchronisationszeichen am Tonkopf vorbeilaufen, erscheint entweder Muster ② (unwahrscheinlich) oder Muster ① auf den Displays. Jetzt kan die eigentliche Einstellung beginnen.
6. Mehrgangpoti P1 ist so einzustellen, daß die Displaysegmente ausschließlich so aufleuchten, wie in Bild 7 unter ② angegeben ist. Falls das nicht gleich gelingt, wird das Band zurückgespult und noch einmal von vorn abgespielt. Das Leseprogramm kann dabei eingeschaltet bleiben. Bei richtiger Einstellung von P1 ist während der Wiedergabe der Synchronisationszeichen ständig Muster ② aus Bild 7 sichtbar; auch kurzzeitig darf die Anzeige nicht zu einem anderen Muster überwechseln.
7. Die Daten (in diesem Fall Synchronisationszeichen) werden innerhalb eines bestimmten Einstellbereichs von P1 korrekt vom Band gelesen. Die richtige Einstellung liegt genau in der Mitte dieses Bereichs. Wenn man

also die Potischraube nach links ebenso weit wie nach rechts drehen muß, damit statt des Musters ② das Muster ① erscheint, ist der endgültige Punkt gefunden.
Dies war die erste Methode, mit der das "Auge" (oder treffender gesagt das "Ohr") des Junior-Computers geschärft werden kann. Zur Sicherheit setzt man ein Programm, von dem ein Listing vorhanden ist, probeweise auf Band und schaltet den Computer danach eine Zeit lang aus. Wenn das RAM seinen Inhalt verloren hat, liest man das Programm wieder ein und vergleicht es Byte für Byte oder Instruktion für Instruktion mit dem auf Papier festgehaltenen Original.

2. PLL-Einstellung mit Oszilloskop

Mit einem Oszilloskop (eine Zweikanalausführung ist vorteilhaft, aber nicht unbedingt notwendig) läßt sich das PLL-Ausgangssignal unmittelbar sichtbar machen, so daß der Umweg über ein Lese-Hilfsprogramm entbehrlich ist. Zuerst wird ein Testsignal auf Band gesetzt, das nur aus Nullen und Einsen besteht: Ø, 1, Ø, 1, Ø, 1 usw. Dieses Testprogramm ist in Tabelle 3 angegeben. Der Y-Eingang des Oszilloskops liegt am PLL-Ausgang; dies ist Pin 7 von IC7 oder der Verbindungspunkt R30/R31 in der Schaltung Bild 2 aus Kapitel 10. Wenn P1 richtig eingestellt ist, zeigt der Oszilloskopschirm ein PLL-Ausgangssignal wie in Bild 12c dargestellt und vom Foto in Bild 15 wiedergegeben. Die Folgen einer Fehleinstellung von P1 sind dagegen in Bild 12a und Foto Bild 13 sowie in Bild 12b und Foto Bild 14 festgehalten. In allen Fällen arbeitet der PLL; er ist "eingerastet", weil P1 ungefähr in Mittelstellung steht.
Die in Bild 12 gezeichneten Signale sind mit den Signalen identisch, die die Bilder 13 . . . 15 zeigen. In den Fotos überlappen sich Signalteile, und zwar immer dann, wenn das Signal von einem zum anderen logischen Zustand springt. Dies liegt an dem schon besprochenen PLL-Jitter sowie an der nicht unendlich kurzen Belichtungszeit bei der Aufnahme der Fotos. Betrachten wir die Bilder 13 . . . 15 genauer, stellen wir fest, daß beim Lesen des nur aus Nullen und Einsen bestehenden Testsignals anscheinend gleichzeitig eine logische Ø und eine logische 1 vom Band kommen. Dies hat jedoch seine Ursache darin, daß Nullen und Einsen vom Oszilloskop übereinandergeschrieben werden.

Tabelle 3. Hilfsprogramm, das abwechselnd Nullen und Einsen auf das Band schreibt. Dieses Bitmuster dient zum Einstellen des PLL mit dem Oszilloskop. Gestoppt wird das Programm durch Drücken der RST-Taste.

```
M
HEXDUMP: 2ØØ,23F
        Ø  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
Ø2ØØ:  A9 7D 8D 6C 1A A9 C3 8D 6D 1A A9 Ø3 8D 76 1A A9
Ø21Ø:  Ø2 8D 77 1A A9 47 A2 FF 8D 82 1A 8D 78 1A 8E 83
Ø22Ø:  1A A9 ØØ A2 7F 8D 8Ø 1A 8E 81 1A 2Ø C8 ØA 2Ø E5
Ø23Ø:  ØA 2Ø E5 ØA 2Ø C8 ØA 2Ø C8 ØA 2Ø E5 ØA 4C 2B Ø2
Ø24Ø:

JUNIOR
```

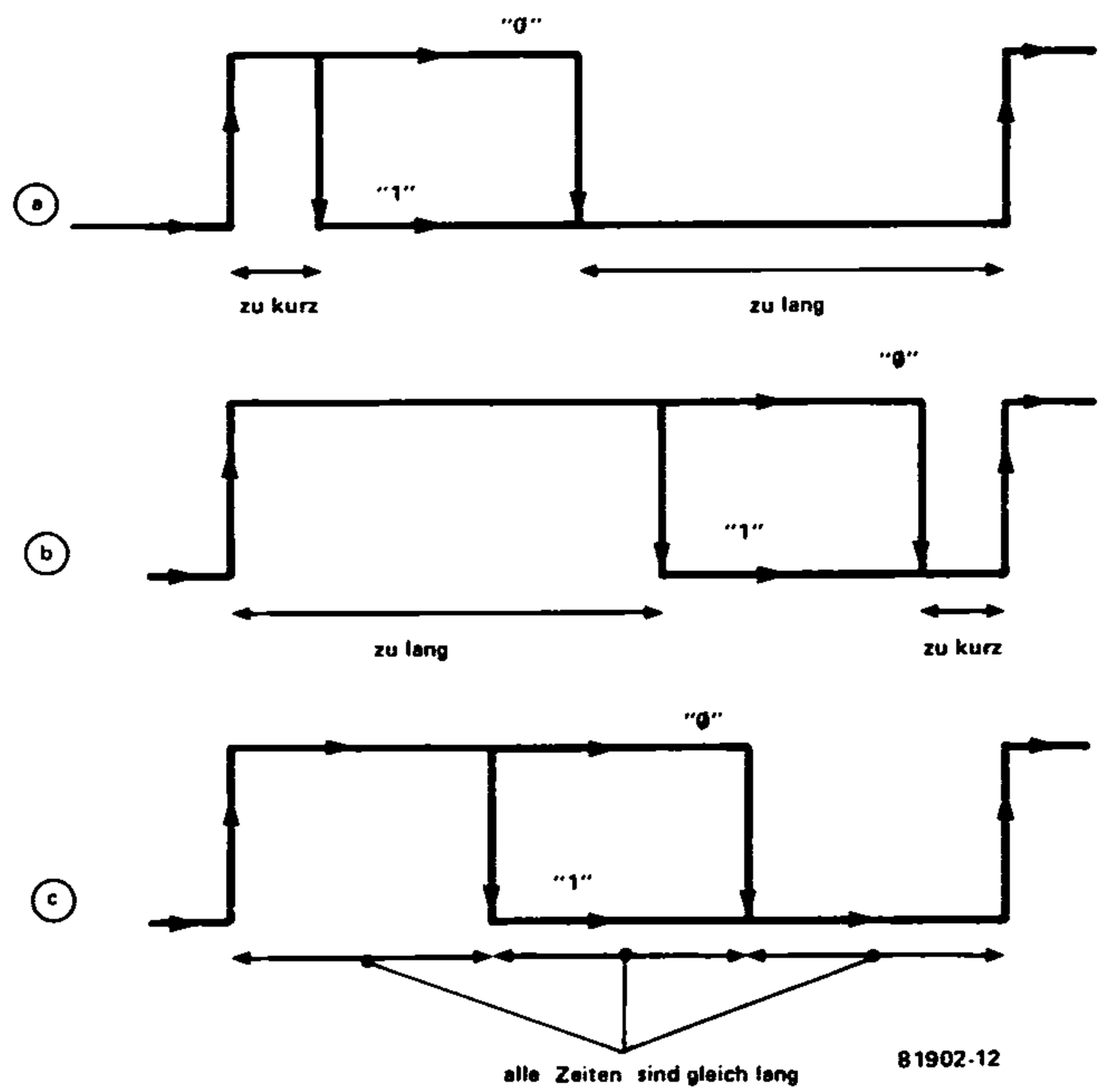


Bild 12. Das Ausgangssignal des eingerasteten PLL bei den logischen Signalen Ø und 1. In den Fällen ⓐ und ⓑ ist Poti P1 auf der Interfacekarte falsch eingestellt, während im Fall ⓒ alles zum besten steht.

Auf den Fotos entsprechen einem Bit neun Skalenteile; der Abbildungsmaßstab hängt natürlich von der Einstellung der Zeitbasis des Oszilloskops ab. Bei allen Signalen in Bild 12 ist das Signal für logisch Ø länger "hoch" und deshalb kürzer "niedrig" als das Signal für logisch 1. Dies steht in Einklang damit, daß zur logischen Ø ein längerer Anteil des höherfrequenten Signals gehört als zur logischen 1. Bild 6a hat dies bereits verdeutlicht.
Die in Bild 12 gezeichneten Signale unterscheiden sich in der Länge der drei Signalabschnitte: In Bild 12a ist die Zeit, in der das Signal "hoch" ist, zu kurz; die Zeit, in der das Signal "niedrig" ist, dauert dagegen zu lange. Das gilt sowohl für logisch Ø als auch für logisch 1. In Bild 12b liegen die Zeitverhältnisse genau umgekehrt. Die Verschiebungen in die eine oder andere Richtung kommen durch fehlerhafte Einstellungen von Poti P1 zustande. Einmal ist die Potischraube zu weit nach links verdreht, im anderen Fall steht sie zu weit rechts. Bild 12c und Foto 15 zeigen das Signal bei korrekter Einstellung: Die Zeit, während der das Signal sowohl bei logisch Ø als auch bei logisch 1 "hoch" ist, stimmt mit der Zeit überein, während der das Signal unabhängig vom logischen Zustand "niedrig" ist. Ferner hat auch der mittlere Signalabschnitt, der bei logisch Ø "hoch" und bei logisch 1 "niedrig" ist, die gleiche Länge.
Soviel zur Einstellung des PLL mit dem Oszilloskop. Zum Schluß dieses Abschnitts wollen wir noch die Frage klären, weshalb trotz des weiten PLL-Fangbereichs eine präzise Einstellung von Poti P1 notwendig ist. In

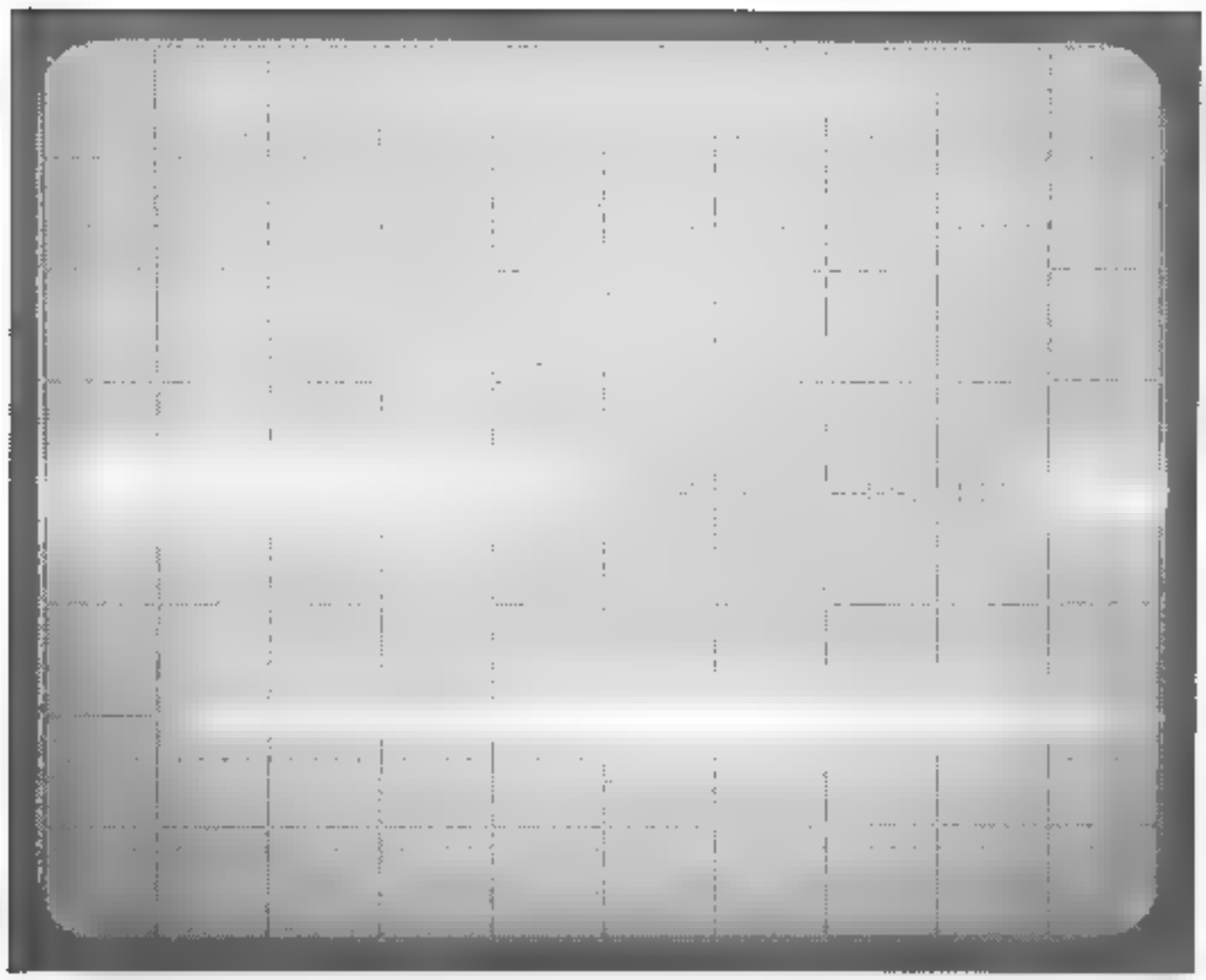


Bild 13. Vergleiche Bild 12a: nicht oder nicht korrekt eingestellter PLL.

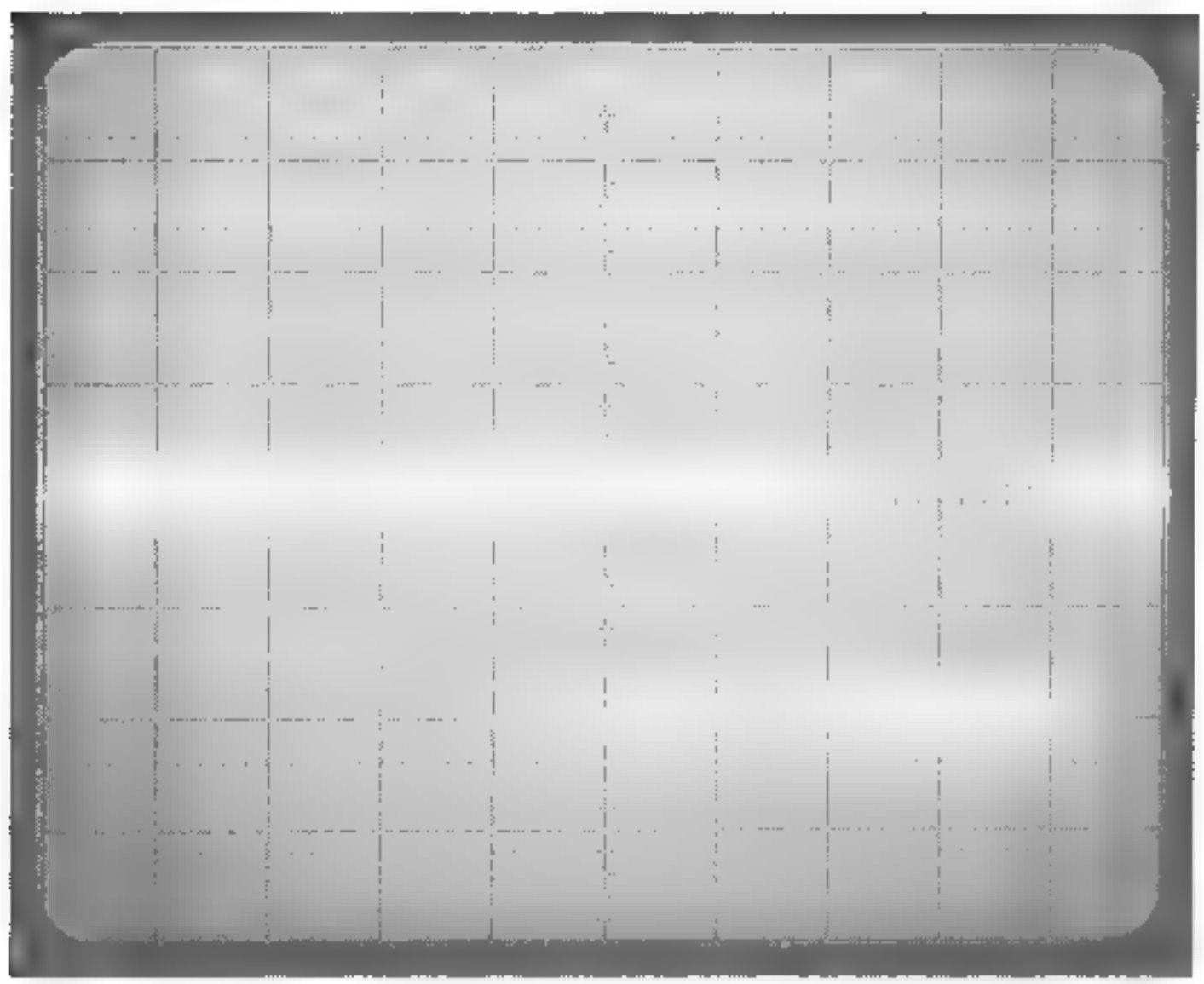


Bild 14. Vergleiche Bild 12b: nicht oder nicht korrekt eingestellter PLL, Poti P1 jedoch in Gegenrichtung verdreht.

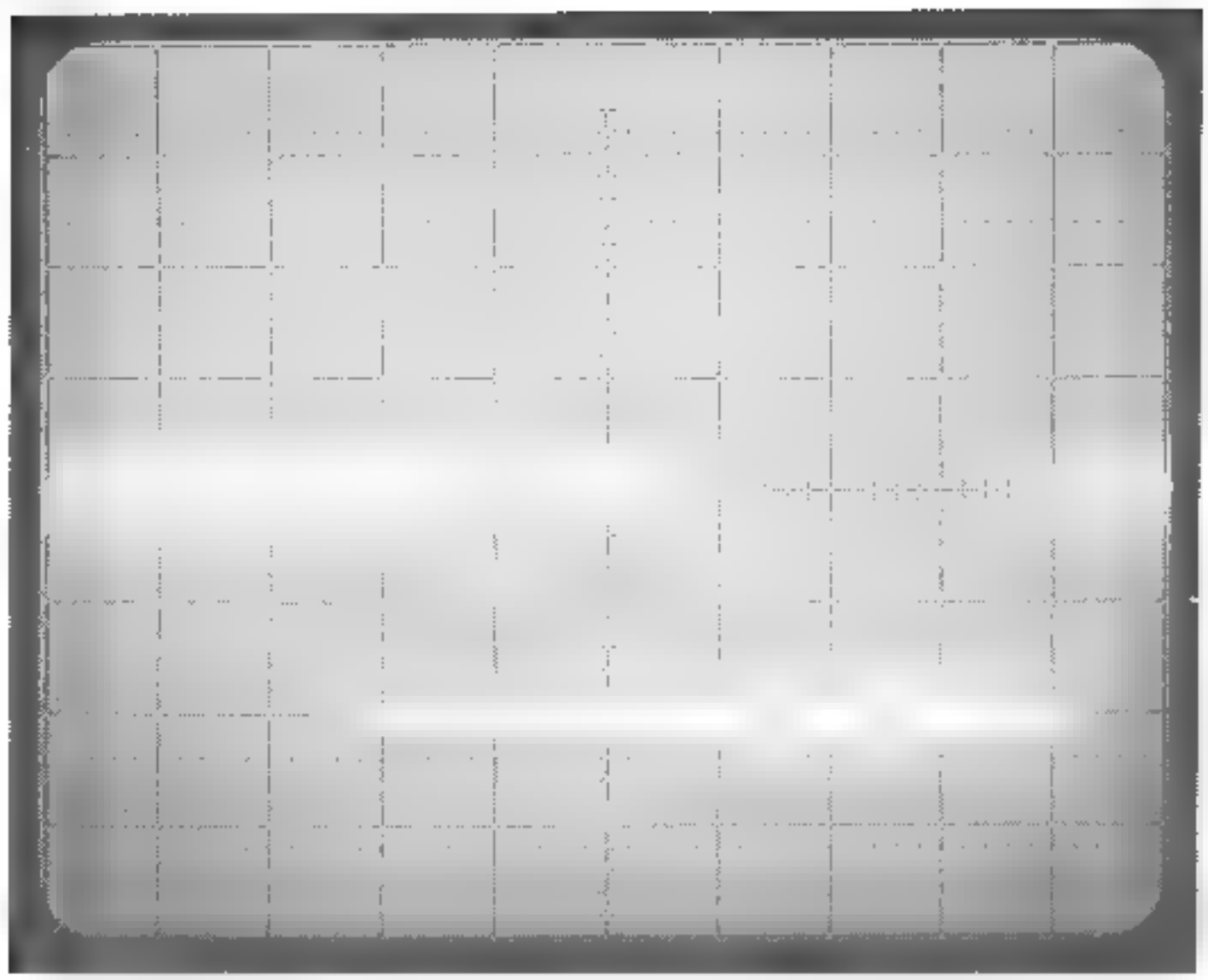


Bild 15. Vergleiche Bild 12c: korrekt eingestellter PLL.

Bild 16 sind die VCO-Leerlauffrequenz und die auf dem Band stehenden Tonfrequenzen als Pfeile dargestellt. Bild 16a zeigt die Lage der Frequenzen bei korrekt eingestelltem PLL, während die Bilder 16b und 16c zu einem einwandfrei arbeitenden, jedoch falsch eingestellten PLL gehören. Nicht funktionsfähig ist der PLL, wenn Poti P1 rechts oder links am Anschlag steht; dann liegen die Signalfrequenzen wie in Bild 16d und 16e skizziert.
Bild 16f, das Ähnlichkeit mit 16b hat, gibt die Situation bei Verwendung von zwei Kassettenrekordern mit unterschiedlicher absoluter Bandgeschwindigkeit wieder. Die Datensendungen werden von Rekorder A aufgenommen und von Rekorder B wiedergegeben, wobei die Geschwindigkeit von B um 10% höher als die Geschwindigkeit von A ist. Umgekehrt liegen die Verhältnisse in Bild 16g: Hier läuft Rekorder B um 10% langsamer als Rekorder A. In der Praxis sind derart große Abweichungen zwar recht selten (nominal beträgt die Bandgeschwindigkeit 4,76 cm/s), doch an diesen Extremfällen wird das Problem besonders leicht deutlich. Eigentlich müßte die Leerlauffrequenz des PLL-VCO für die Aufnahme anders eingestellt sein als für die Wiedergabe; das deuten die durchbrochen gezeichneten Pfeile in Bild 16f und 16g an. Da ein ständiges Nachstellen in der Praxis ziemlich umständlich wäre, wird die VCO-Frequenz exakt auf ihren Sollwert abgeglichen, so daß Differenzen der vom Band kommenden Tonfrequenzen mit Sicherheit in den Toleranzbereich fallen.
Zur Veranschaulichung des Ganzen sind noch einige Fotos hinzugefügt: Bild 17 zeigt das Ausgangssignal des Kassetteninterface (Poti P2 voll aufgedreht), wie es beim abwechselnden Schreiben von Nullen und Einsen mit dem Hilfsprogramm aus Tabelle 3 entsteht. Deutlich sind die Teil-

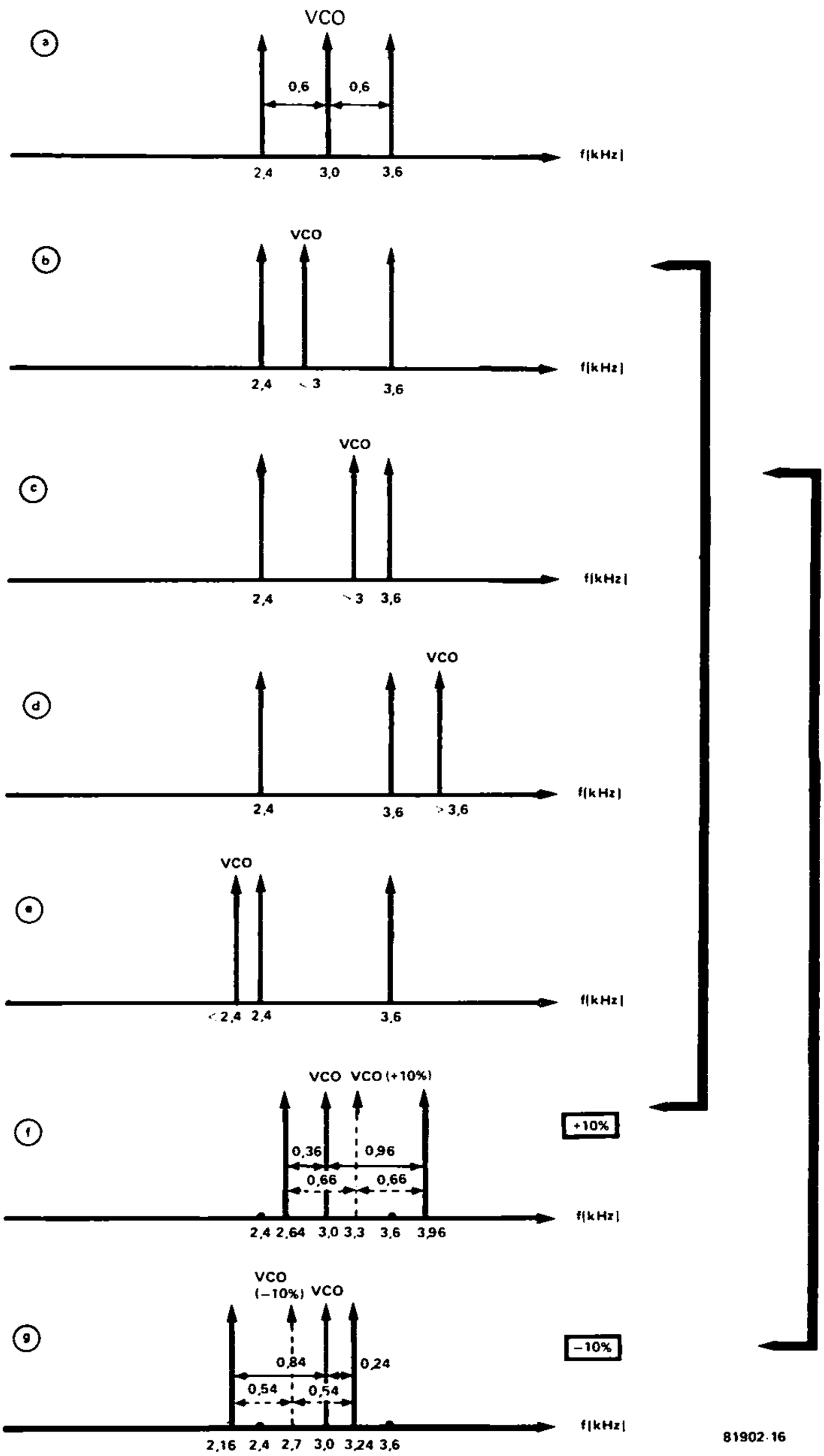


Bild 16. Hier sind siebenmal jeweils drei Frequenzen durch Pfeile dargestellt: die hohe und die niedrige Tonfrequenz der Bit auf dem Band (vergleiche Bild 5a) sowie die VCO-Frequenz des PLL, die mit P1 einstellbar ist.

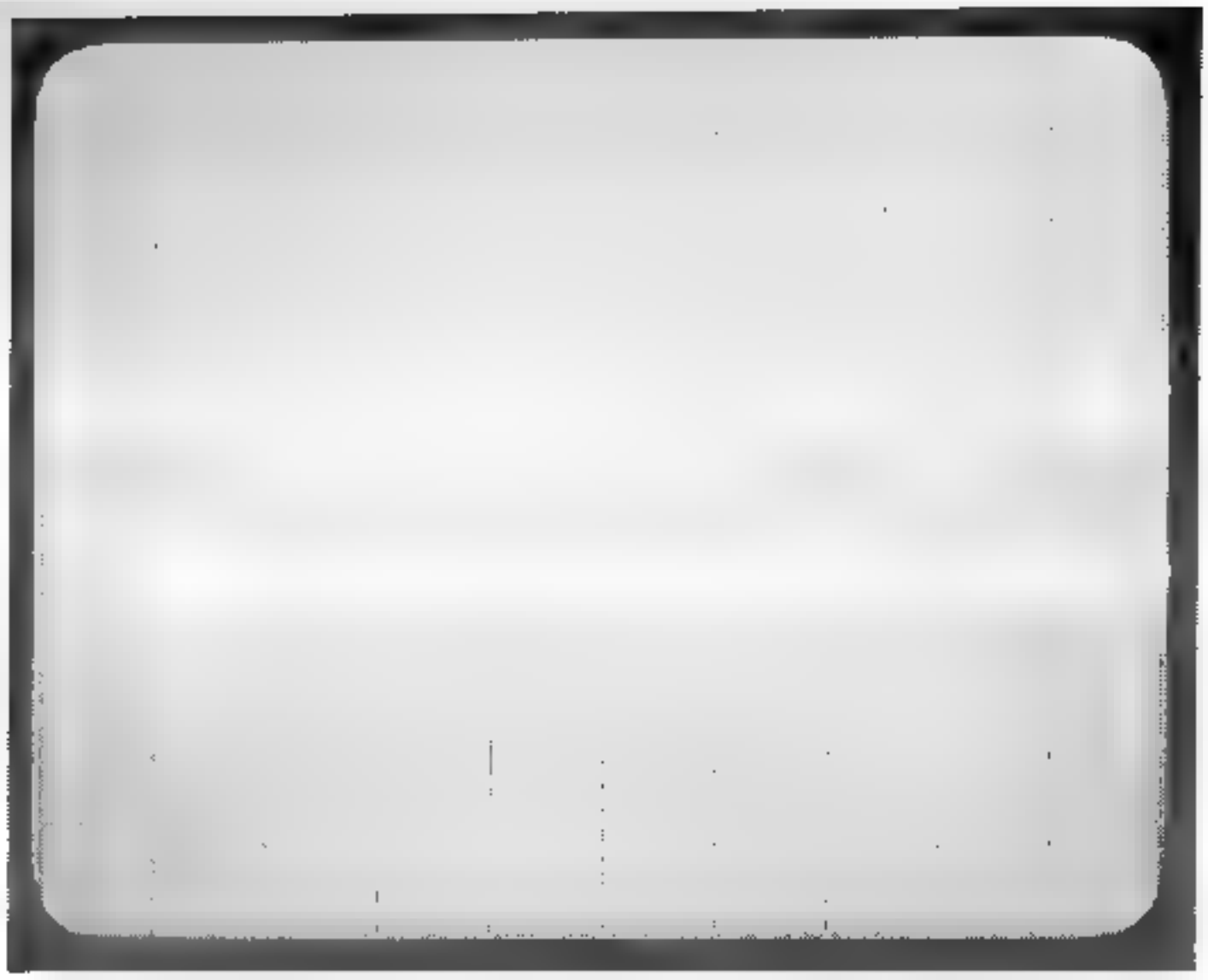


Bild 17. So sieht das Ausgangssignal vom Testprogramm (Tabelle 3) am Anschluß J2 des Kassetteninterface aus. Dabei ist der Pegeleinsteller P2 ganz aufgedreht. Die Vertikaleinstellung des Oszilloskops ist 2 V/Teilung.

Bild 18. Die Einsen und Nullen aus dem Testprogramm in Tabelle 3 gibt der Kassettenrekorder in Form der oberen Signalfolge an den Eingang J1 des Kassetteninterface ab. Dabei ist am Kassettenrekorder die Lautstärke ganz aufgedreht. Das untere Signal entspricht der PLL-Ausgangsspannung. Die Einstellung für beide Signale entspricht auch hier 2 V/Teilung.

Bild 19. Die Signalinhalte sind mit denen aus Bild 18 identisch. Der Lautstärkeeinsteller des Kassettenrekorders ist allerdings nur ganz minimal aufgedreht. Die PLL-Schaltung arbeitet jedoch auch noch bei dieser minimalen Signalamplitude einwandfrei. Um jedoch eventuelle Störungen auf ein Mindestmaß zu reduzieren, sollte der Amplitudenwert 4 V nicht unterschreiten.

abschnitte der hohen Frequenz (ca. drei halbe Perioden pro Skalenteil) und der niedrigen Frequenz (ca. eine ganze Periode pro Skalenteil) erkennbar. Der Unterschied zwischen dem Signal für logisch Ø und dem Signal für logisch 1 ist ebenfalls offensichtlich. Die Oszilloskopaufnahme in Bild 18 zeigt oben das Ausgangssignal des Kassettenrekorders in Stellung Wiedergabe (gemessen an Buchse J1), während unten das zugehörige PLL-Ausgangssignal zu sehen ist. Die Aufnahme-Wiedergabe-Kette verformt zwar die Rechtecksignale aus Bild 17 zu einem sinusähnlichen Signal, was jedoch für die Funktion des PLL ohne Bedeutung ist.
Auch Amplitudenunterschiede können dem PLL nichts anhaben, wenn sie innerhalb der relativ weiten Grenzen bleiben: In Bild 18 ist der Lautstärkeeinsteller des Kassettenrekorders voll aufgedreht, in Bild 19 wurde die Signalstärke drastisch reduziert.

Und schließlich . . .

noch einige Winke für die Praxis

1. Wahl des Kassettenrekorders

Der billigste Rekorder ist für die Datenspeicherung auf Band ebenso wenig zu empfehlen wie der teuerste. Während beim ersten wahrscheinlich die elektrische und mechanische Qualität für diesen Zweck kaum ausreicht, tut man im zweiten Fall des Guten etwas zu viel. Ein Monorekorder genügt natürlich; er sollte allerdings unbedingt mit einem Bandzählwerk ausgestattet sein. Ferner ist ein Lautsprecherausgang mit Abschaltung des eingebauten Lautsprechers überaus nützlich, denn er liefert eine hohe, mit dem Lautstärkeknopf einstellbare Ausgangsspannung. Bei vielen Rekorder-Modellen wird der externe Lautsprecher über einen 3,5-mm-Klinkenstecker angeschlossen. Ein Fernbedienungsanschluß ist nur erforderlich, wenn für Aufnahme und Wiedergabe (INPUT und OUTPUT) zwei getrennte Kassettenrekorder vorgesehen sind. In die Fernbedienungsbuchse paßt meistens ein 2,5-mm-Klinkenstecker.

2. Anschluß bei nur einem Rekorder

Man verbindet den Aufnahme-Eingang (häufig eine DIN-Buchse; Kontakt 3 = Signal, Kontakt 2 = Masse) mit J2 und den Lautsprecherausgang (3,5-mm-Klinkenbuchse oder DIN-Lautsprecherbuchse) mit J1. Eine eventuell vorhandene Fernbedienungsbuchse wird nicht benutzt.
Anmerkung: Auch wenn der gleiche Rekorder die Daten aufnimmt und wiedergibt, kann dieser auf Wunsch vom Junior-Computer über die Fernbedienungsbuchse gestartet und gestoppt werden. Dazu müssen die Kontakte von J3 und J4 miteinander verbunden werden; ferner müssen auf der Interfacekarte die beiden Relais (Re1 und Re2) vorhanden sein.

3. Zwei Kassettenrekorder

Während der Kassettenrekorder OUTPUT die Daten auf Band schreibt, werden diese vom Kassettenrekorder INPUT gelesen. Zwischen beiden Rekordern muß deshalb ein reger Kassettenaustausch stattfinden. Die Steuerung vom Computer über die Fernbedienungsbuchsen ist hier sinnvoll und nützlich. Zwischen dem Junior und den Rekordern sind vier Leitungen zu legen: vom Lautsprecherausgang INPUT nach J1, vom Aufnahme-Eingang OUTPUT nach J2, vom Fernbedienungseingang INPUT nach J3 und vom Fernbedienungseingang OUTPUT nach J4. Der Rekorder INPUT bleibt ständig auf Wiedergabe (Play), der Rekorder OUTPUT ständig auf Aufnahme (Record) geschaltet.
Um den PLL bei Verwendung von zwei Kassettenrekordern richtig einzustellen, läßt man zuerst die Testdaten mit dem Hilfsprogramm in Tabelle 2 (erste Hälfte) oder Tabelle 3 vom OUTPUT-Rekorder auf Band setzen. Anschließend wird die Kassette in den INPUT-Rekorder eingelegt und der PLL nach einer der beiden beschriebenen Methoden eingestellt. Die Einstellung berücksichtigt dann auch eine eventuelle Differenz zwischen den Bandgeschwindigkeiten der beiden Kassettenrekorder.
Während der Betriebs startet der betreffende Rekorder sofort nach

Drücken von SAVE, GET oder SEF. Insbesondere bei SAVE und SEF muß das Band möglichst schnell seine Endgeschwindigkeit erreichen, da sonst die ersten Tonsignale bei der Wiedergabe höhere Frequenzen als die Ursprungssignale haben. Beträgt die Bandgeschwindigkeit kurz nach dem Start beispielsweise nur die Hälfte der Sollgeschwindigkeit, werden die in diesem Moment aufgenommenen Signalfrequenzen 2400 Hz und 3600 Hz später mit 4800 Hz bzw. 7200 Hz wiedergegeben. Der PLL ist damit natürlich überfordert. Weil jede Datensendung mit 255 Synchronisationszeichen (Dauer ca. 2,5 s) beginnt, von denen 50 bis 100 fehlen dürfen, sind aber Schwierigkeiten infolge des trägen Bandanlaufs normalerweise nicht zu erwarten.

4. Aufnahmen mit Niveau

Wichtig ist auch, daß die Datensendungen sowohl mit ausreichender Lautstärke auf dem Band stehen (Aussteuerung des Rekorders!) als auch mit genügender Lautstärke wiedergegeben werden. Das spielt deshalb eine große Rolle, weil die Amplitude des am PLL-Eingang ankommenden Signals aus verschiedenen Gründen kurzzeitig oder länger abfallen kann. Der PLL ist zwar, was eine konstante Signalstärke betrifft, alles andere als anspruchsvoll (vergleiche Bild 18 und Bild 19), doch auch hier gibt es eine Grenze. Wird sie unterschritten, dann verschwimmen sozusagen dem PLL die Daten vor Augen.

An Beeinträchtigungen der Lautstärke können vor allem ein nach oben abfallender Frequenzgang des Rekorders sowie die berüchtigten Dropouts schuld sein. Im ersten Fall werden die 3600-Hz-Töne leiser wiedergegeben als die 2400-Hz-Töne. Als Ursache kommen ein verschmutzter, falsch eingestellter oder verschlissener Tonkopf sowie eine zu geringe Aussteuerung bei der Aufnahme in Frage. Die aus der Aufnahme und der Wiedergabe resultierende Gesamtfrequenzcharakteristik benachteiligt nämlich die hohen Frequenzen um so stärker, je schwächer das Band ausgesteuert wurde.

Sogenannte Dropouts entstehen, wenn das Band kurzzeitig nicht vorschriftsmäßig am Tonkopf anliegt, zum Beispiel infolge von Verschmutzungen auf dem Band. Die Lautstärke sinkt dann drastisch ab, trotzdem werden die Schwankungen wegen ihres kurzzeitigen Charakters meistens vom Ohr nicht wahrgenommen. Inzwischen können jedoch zehn oder mehr Bit den Tonkopf passiert haben. Ein totaler Signalausfall tritt glücklicherweise äußerst selten auf; in aller Regel wird das Signal nur um den Faktor 10 . . . 20 (20 . . . 26 dB) geschwächt. Daraus ergibt sich für die Wiedergabelautstärke eine wichtige Forderung: **Das am PLL-Eingang liegende Signal muß eine wesentlich höhere Amplitude haben als unter normalen Umständen notwendig ist.** Wie in Bild 18 oben sichtbar, kann die Eingangsspannung ohne weiteres 5 . . . 6 V_{SS} betragen; das PLL-IC wird hierdurch nicht beschädigt! Das ist auch der Grund dafür, weshalb man das Signal am Lautsprecherausgang abnehmen und den Lautstärkeeinsteller nicht zu knapp aufdrehen sollte. Falls der Rekorder einen Ausgang besitzt, der zum Anschluß eines Verstärkers vorgesehen ist ("Line"), liegt an diesem eine wesentlich niedrigere Signalspannung. **Wird trotzdem ein "Line"-Ausgang benutzt, dann entfällt Widerstand R27 auf der Interfaceplatine!** Wenn der Kassettenrekorder mit einer Höhen-Klangeinstellung

ausgestattet ist, und der Computer am Lautsprecherausgang liegt, stellt man den Ton möglichst hell ein.
Die Aussteuerung des Bandes während der Aufnahme darf ebenfalls nicht vernachlässigt werden. Auch wenn die Über-Band-Frequenzcharakteristik des Rekorders im interessierenden Bereich geradlinig verläuft, kann eine zu schwache Aussteuerung bei der Aufnahme nur begrenzt durch eine höhere Wiedergabelautstärke ausgeglichen werden. Trimmpoti P2 auf der Interfacekarte darf ruhig auf Rechtsanschlag stehen (siehe Bild 17; die Signalspannung beträgt dort mehr als 4,5 V_{SS}), insbesondere beim Anschluß eines Rekorders mit Aussteuerungsautomatik.
Wie schon erwähnt, wirkt sich ein verschmutzter Tonkopf nachteilig auf die Datenaufnahme und -wiedergabe aus. Der Tonkopf ist deshalb von Zeit zu Zeit zu reinigen. Kassetten werden heute in den unterschiedlichsten Preislagen angeboten. Sparsamkeit auf Kosten der elektrischen und mechanischen Qualität ist fehl am Platz. Die Lebensdauer der Kassetten ist selbstverständlich nicht unbegrenzt, so daß bei häufigem Gebrauch von Zeit zu Zeit Neuanschaffungen fällig sind. Eine gute Idee ist das Aufzeichnen des gleichen Programms auf zwei Kassetten; es steht dann immer eine Reservekopie zur Verfügung. Oder von den Kassetten wird nach einer bestimmten, angemessenen Zahl von Durchläufen ein Duplikat angefertigt. Danach benutzt man nur noch das Duplikat, während die ältere Kassette für weniger kritische Aufnahmen (Sprache, Unterhaltungsmusik) weiterverwendet werden kann.

Wir haben nun alles erfahren, was es an Wissenswertem über das Speichern von Daten auf Magnetband in Zusammenhang mit dem Junior-Computer zu berichten gibt. Dazu gehörte die Aufzeichnungsmethode, die Gebrauchsanweisung für TM und das Einstellen der PLL-Schaltung. Natürlich durften einige Ratschläge aus der Praxis für die Praxis nicht fehlen. Worüber wir bisher noch nichts sagten, ist die erste Inbetriebnahme der Kassettenhardware und -software nach der Fertigstellung des erweiterten Junior-Computers. Damit setzen wir nachfolgend den Schlußpunkt dieses Kapitels.

5. Läuft er, ist es gut!

Läuft er nicht, helfen wir ihm auf die Beine

Wir gehen davon aus, daß die Hardware des erweiterten Junior-Computers nach seiner Fertigstellung entsprechend den Hinweisen in Kapitel 10 kontrolliert wurde. Deshalb folgt jetzt die Überprüfung der Software.
Arbeitet das Systemprogramm TM? Diese Frage läßt sich weitgehend ohne Anschluß eines Kassettenrekorders klären. Voraussetzung ist, daß ein mit TM programmiertes EPROM des Typs 2716 in der Fassung für IC4 (**nicht IC5!**) steckt, daß das EPROM vorschriftsmäßig eingesetzt wurde und daß alle 24 Pins Kontakt haben. Wir geben nun AD Ø 8 1 Ø GO ein (Startadresse von TM ist Ø81Ø, **nicht Ø8ØØ**!). Auf den Displays muß "id ØØ" erscheinen. Drückt man EDIT, dann zeigen die Displays "77" an (ENDAD höher als BEGAD). Beim Drücken von SAVE leuchtet die rote OUTPUT-LED D5 auf; sofern Relais Re2 vorhanden ist, sind die Kontakte von J4 miteinander verbunden (auf Durchgang prüfen!). Der Junior-Computer schickt auch ohne angeschlossene(n) Kassettenrekorder Datensendungen nach außen:

EA wird höher als SA gewählt und anschließend Taste SAVE gedrückt. Es muß die Rückmeldung "id XX" folgen. Ebenso läßt sich die Funktion von SEF kontrollieren. Man setzt einfach mit dem Editor ein Phantasieprogramm in das RAM, wobei die Spielregeln für BEGAD und ENDAD natürlich beachtet werden müssen. Danach wird SEF gedrückt. Folgen muß wieder die Rückmeldung "id XX", die rote OUTPUT-LED muß aufleuchten und die Kontakte von J4 müssen miteinander verbunden sein.
Was ist zu tun, wenn sich hier ein Fehler zeigt? Man kontrolliert zuerst den Inhalt des TM-EPROMs an Hand des Hexdump, daß im Anhang 4 steht (RST AD Ø 8 Ø Ø und dann fortlaufend + drücken). Dann wird die Hardware in der Umgebung von T3 überprüft. Sind die fünf Leitungen zwischen der Interfacekarte und dem Portkonnektor der Basiskarte vorhanden und richtig angeschlossen?
Ohne Kassettenrekorder können wir nicht nur "trocken" schreiben, sondern auch lesen. Nach Drücken von GET muß die grüne INPUT-LED aufleuchten, die Kontakte von J3 müssen miteinander verbunden sein (sofern Relais Re1 vorhanden ist), und Muster ① aus Bild 7 muß auf den Displays erscheinen, zumindest dann, wenn auf Buchse J1 Störspannungen gegeben werden (mit dem Finger berühren!). Die Displays leuchten wahrscheinlich nur schwach, jedoch ist das abwechselnde Aufleuchten von Di5 und Di6 deutlich sichtbar.
Haben wir bisher keinen Fehler gefunden, dann wird zur Sicherheit noch einmal der PLL eingestellt. Die Einstellvorschriften wurden bereits in einem vorangegangenen Abschnitt dieses Kapitels gegeben. Arbeitet der Junior immer noch nicht wunschgemäß, so führt möglicherweise die Beantwortung folgender Fragen auf die Spur des Fehlers:

- Wurden die Testprogramme richtig eingegeben? (Kontrolle mit dem Monitor!)
- Steht Trimmpoti P2 auf Rechtsanschlag? Dann können natürlich keine Testdaten auf Band geschrieben werden.
- Stand P1 vor der PLL-Einstellung ungefähr in der Mitte? Anderenfalls arbeitet der PLL nicht; er rastet nicht ein (siehe Bild 16d und 16e).
- Reicht die Aussteuerung bei der Aufnahme und die Lautstärke bei der Wiedergabe aus?
- Stimmen die Spannungen an den Meßpunkten der PLL-Hardware (IC6, IC7 und Umgebung) mit den in Kapitel 10, Bild 18 angegebenen Sollwerten überein?
- Wenn am INPUT-Rekorder der "Line"-Ausgang benutzt wird: Ist Widerstand R37 auf der Interfacekarte vorhanden? Er muß in diesem Fall entfernt werden.
- Steht ein Oszilloskop zur Verfügung, dann werden die Signale aus den Bildern 13 . . . 15 und 17 . . . 19 sichtbar gemacht und mit diesen Bildern verglichen. Auf diese Weise läßt sich der Fehler immer lokalisieren.

Die vorstehenden Anweisungen zur Fehlersuche konnten nur von allgemeiner Art sein. Wenn der Erfolg trotz aller Bemühungen ausbleibt und sich der Junior-Computer hartnäckig weigert, seinen Dienst aufzunehmen, ist weiterer guter Rat nicht teuer: Der technische Leserservice der Zeitschrift "Elektor" steht auch allen Lesern dieses Buches zur Verfügung. Wie

man ihn in Anspruch nimmt, darüber gibt die jeweils aktuelle Ausgabe dieser Zeitschrift Auskunft.

Endkontrolle

In der Regel wird eine Fehlersuche nicht notwendig sein, so daß das Zusammenspiel von Kassettenhard- und -software sofort praktisch erprobt werden kann. Wie schon an anderer Stelle in diesem Kapitel beschrieben, kann ein Testprogramm auf Band gesetzt werden, von dem ein Listing oder ein Hexdump existiert. Nach vorübergehendem Ausschalten des Junior-Computers wird dieses Programm vom Band gelesen und Byte für Byte auf seinen Originalzustand kontrolliert. Um ganz sicher zu gehen, kann der betreffende RAM-Abschnitt vor dem Wiedereinlesen zum Beispiel mit Nullen überschrieben werden. Es ist nämlich schon vorgekommen, daß der RAM-Inhalt auch nach Abschalten des Computers während eines ganzen Wochenendes noch fast komplett war!

Schreib- und Lesefehler

Die Funktion der Kontrollbytes CHKL und CHKH (siehe Bild 3 und 4) wurde in diesem Kapitel ausführlich behandelt. Es bleibt noch die Frage offen, was bei Unstimmigkeiten zwischen den auf Band gesetzten und den beim Lesen ermittelten Werten von CHKL und CHKH geschieht. Übrigens: Verglichen werden die Werte nur für die jeweils gesuchte und gefundene Datensendung.
Nehmen wir an, es wurde GET gedrückt. Der Rekorder spielt die eingelesene Kassette ab. Ist die gesuchte Datensendung gefunden, erscheint nach Muster ② aus Bild 7 (Synchronisationszeichen) das Muster ③ auf den Displays: der Datenblock wird in das RAM des Junior-Computers geschrieben. Nachdem dies erledigt ist und sowohl CHKH als auch CHKL stimmen, meldet sich der Computer mit "id XX" zurück (XX = Programmnummer). Die Rückmeldung zeigt an, daß er die Subroutine RDTAPE verlassen hat und zum Hauptprogramm von TM zurückgesprungen ist.
Stimmt CHKH oder CHKL nicht mit den auf Band stehenden Werten überein, so verbleibt der Computer in der Subroutine RDTAPE; eine Rückmeldung "id XX" erfolgt dann nicht. Die Displays wechseln von Muster ③ aus Bild 7 nach Muster ① und danach eventuell nach Muster ② über: Den Tonkopf passieren die Synchronisationszeichen der folgenden Datensendung (die bei ID = FF in das RAM gesetzt wird).
Der Wechsel von Muster ③ nach Muster ① anstelle von "id XX" ist ein Signal dafür, daß beim Lesen der Datensendung vom Band höchstwahrscheinlich ein Übertragungsfehler aufgetreten ist. Höchstwahrscheinlich deshalb, weil, wie schon erklärt, die angewendete Kontrollmethode keine absolute Sicherheit bietet. Man kann in diesem Fall das Band zurückspulen und die Datensendung ein zweites Mal einlesen. Bleibt die Rückmeldung "id XX" wieder aus, kommen folgende Ursachen in Betracht: a) zu niedrige Aufnahme- oder Wiedergabelautstärke; b) richtige Aufnahme- und Wiedergabelautstärke, aber zu viele Dropouts, von denen einige zu stark sind (diese Ursache ist außerordentlich selten, jedoch nicht völlig auszuschließen); c) die Bandgeschwindigkeit des aufnehmenden (fremden) Kassettenrekorders weicht erheblich von der Bandgeschwindigkeit des

wiedergebenden (eigenen) Kassettenrekorders ab. Gemeint ist nicht die Geschwindigkeitsdifferenz zwischen INPUT- und OUTPUT-Rekorder (sie wurde bei der PLL-Einstellung berücksichtigt), sondern der Unterschied, der bei geliehener oder gekaufter Software zu Tage treten kann.
Eine sichere Methode, mit der sich solche Differenzen unschädlich machen lassen, ist folgende: An den Anfang der für fremden Gebrauch bestimmten Kassetten werden Testdaten gesetzt, zum Beispiel vier Minuten lang Synchronisationszeichen mit dem Programm aus Tabelle 2 (erste Hälfte). Die PLL-Schaltung des lesenden Computers läßt sich später mit Hilfe des Programms aus Tabelle 2 (zweite Hälfte) exakt einstellen.
Derartige Vorsichtsmaßregeln kann man zwar treffen; ob sie in der Praxis notwendig sind, ist jedoch eine andere Frage. Er scheint eher so, als ob hier Panzerglasscheiben zur Abwehr von Mücken eingesetzt werden . . .

Das 11. Kapitel ist damit beendet. Es wurde beschrieben, wie der Junior-Computer Daten auf Band schreibt, und man konnte lesen, wie er sie vom Band liest. Nach der Lektüre von dem, was in diesem Kapitel geschrieben steht, können wir das Lesen und das Schreiben von Daten ohne Besorgnis dem Junior-Computer und den Kassettenrekordern überlassen.

12

Junior auf dem Bildschirm

Mehr Sicht- und Tastbares

Über das Tastenfeld des "Junior-Junior-Computers" lassen sich Daten eingeben und alle notwendigen Funktionen steuern, während das sechstellige Siebensegmentdisplay die eigenen Aktionen oder die Aktionen des Computers sichtbar macht. Diese verhältnismäßig einfache Ein- und Ausgabeeinheit ist überraschend vielseitig. Doch leider kann sie nicht alle wünschenswerten Funktionen vereinen und nur relativ wenig Bedienungskomfort bieten. Deshalb haben wir die Möglichkeit geschaffen, ein Datensichtgerät (ein Terminal) an den Junior anzuschließen. Erst damit wird der Junior endgültig zum professionellen Senior.
Das Elekterminal, das Videointerface mit ASCII-Tastatur, wird mit dem Junior-Computer auf der einen Seite und einem TV-Gerät auf der anderen Seite verbunden. Während die LED-Displays nur eine einzige "Zeile" gleichzeitig sichtbar machen konnten, bietet der Bildschirm bei weitem mehr Raum. Mit der ASCII-Tastatur ist eine bequeme Eingabe von Daten und auch die Steuerung grafischer Funktionen möglich.
Auch hier arbeiten wieder Hard- und Software Hand in Hand: Ohne das Systemprogramm Printer Monitor, einer erweiterten Version des Standard-Monitors, wäre das Terminal nutzlos. An seiner Stelle läßt sich übrigens auch ein Drucker oder ein anderes peripheres Gerät anschließen, das Daten empfangen oder senden kann. Damit ist es nun möglich, ein "Junior-Journal" auf den Bildschirm oder zu Papier zu bringen, das nicht nur die

gerade vollzogene Handlung, sondern auch die früheren Geschehnisse festhält. Der "Nachrichtenredakteur" ist das Programm PM; es ist nicht ausgeschlossen, das sei hier noch angemerkt, daß er die Meldungen später in einer höheren Programmiersprache, zum Beispiel BASIC, präsentiert.

Vom Spielzeugpiano zum Konzertflügel und vom Kaleidoskop zum großen TV-Gerät – so könnte man den Übergang zur ASCII-Tastatur und zum Videodisplay umschreiben. Damit soll die Ein- und Ausgabeeinheit der Junior-Computer-Basisversion nicht etwa abgewertet werden, im Gegenteil. Für den Einstieg in die Mikrocomputertechnik ist ein übersichtliches Tastenfeld und eine leicht überschaubare Anzeige besser geeignet als eine verwirrende Vielfalt von Tastenfunktionen und optischen Darstellungsmöglichkeiten. Die Tastatur auf der Basisplatine hat die für den Anfänger richtige Dimension, zumal die fünf Software-Steuertasten seit Kapitel 11 mit drei Funktionen (Standard-Monitor, Editor und Tape Management) belegt sind. Technisch gesehen ließe sich die Anzahl der Funktionen pro Taste leicht ausweiten, doch bei der Bedienung gibt es dann Probleme. Wo soll man die weiteren Tastenbeschriftungen anbringen, vielleicht ähnlich den Inschriften verschiedener Münzen auf dem Rand? Man könnte zwar wie bei einer Schreibmaschine eine Umschalttaste vorsehen oder zusätzliche Steuerfunktionen einer bestimmten Reihenfolge zuordnen, in der drei oder mehr Tasten zu bedienen sind. Der Übersichtlichkeit wäre dies aber sicherlich kaum dienlich. Schlußfolgerung: Anschluß einer großen ASCII-Tastatur, auch im Vorausblick auf das spätere Arbeiten mit einer höheren Programmiersprache.
Visuell waren die Möglichkeiten bisher ebenfalls ziemlich eng begrenzt. Das LED-Display gibt nur eine einzige Adresse mit zugehörigen Daten oder eine Drei-Byte-Instruktion oder sechs andere Daten-Nibbles wieder. Hier sind Erweiterungen unterschiedlicher Art möglich:

1. Mehr Textelemente. Auf einem Siebensegmentdisplay lassen sich neben dezimalen Ziffern nur wenige andere Textelemente wie Groß- und Kleinbuchstaben, Satzzeichen usw. naturgetreu und leserlich darstellen. Der Zeichenvorrat einer Schreibmaschine ist für die Ein- und Ausgabeeinheit eines Computers nicht nur nützlich, sondern bei höheren Programmiersprachen sogar unerläßlich. Dabei kann jedoch auf das Umschalten zwischen Groß- und Kleinbuchstaben sowie auf die Darstellung von Akzentzeichen (', `, ^) verzichtet werden.
2. Mehr Zeichen pro Zeile. Sechs Testelemente (Charakter) pro Zeile sind zweifellos nicht viel. Ein Hexdump bekäme mit dieser Zeilenlänge ein recht unhandliches Format. Beim Elekterminal passen deshalb 64 Zeichen in jede Zeile, vergleichbar mit einem 64-teiligen LED-Display.
3. Mehr Zeilen. Das LED-Display kann immer nur eine einzige Zeile gleichzeitig wiedergeben. Der Text dieser Zeile folgt den Aktionen des Programmierers oder des Programms; die bisherige Zeile wird dabei überschrieben. Zum einen läßt sich der vorangegangene Ablauf nicht mehr nachvollziehen, zum anderen ist die Darstellung mehrerer Zeilen unmöglich, deren Texte das Ergebnis einer durch Tastendruck ausgelösten

Steuerfunktion sind. Das Elekterminal macht dagegen auf dem Bildschirm bis zu 16 Zeilen gleichzeitig sichtbar. Schließt man statt des Terminals einen geeigneten Drucker an, so ist die Anzahl der Zeilen (solange der Papiervorrat reicht) sogar unbegrenzt.
Alle genannten Wünsche, die die Ausweitung der Ein- und Ausgabemöglichkeiten betreffen, werden durch den Anschluß des Elekterminals mit seiner ASCII-Tastatur an das RS-232-Interface des Juniors erfüllt. Was dann noch fehlt, ist ein gewöhnlicher Schwarz-Weiß-TV-Empfänger.
Junior-Computer-Freunden, die seit mindestens Ende 1978 die Zeitschrift "Elektor" lesen, sagen wir nachfolgend nichts Neues: Das Elekterminal mit der ASCII-Tastatur und dem TV-Modulator sind keine speziell für den Junior-Computer entworfenen Einheiten. Sie wurden universell ausgelegt und seinerzeit im Rahmen eines anderen Hobby-Computer-Projekts (SC/MP-System) in "Elektor" veröffentlicht.
Es ist nicht möglich, die betreffenden Zeitschriftenartikel an dieser Stelle noch einmal in "Junior-Version" zu wiederholen. Das würde nämlich ein weiteres Kapitel füllen und deshalb zu viel Platz in Anspruch nehmen. In diesem Kapitel werden jedoch alle Aspekte beleuchtet, die in unmittelbarem Zusammenhang mit dem Junior-Computer stehen. Um das Elekterminal zu bauen und seine Arbeitsweise zu verstehen, müssen die genannten Veröffentlichungen natürlich zur Hand sein. Wenn das nicht der Fall ist, hilft der Elektor-Kopien-Service (EKS) weiter. Wie man die Kopie eines bestimmten Artikels bestellt, ist im jeweils aktuellen Heft von "Elektor" nachzulesen. Eine Alternative für die Bestellung von Kopien bietet sich übrigens in Form des Buchs *"µComputer für den Selbstbau, Teil 2"* an. Dieses Buch enthält neben SC/MP-spezifischen Beschreibungen auch alle Originalartikel zum Elekterminal sowie darüber hinaus noch einiges mehr: einen leicht verständlichen BASIC-Kurs für das Selbststudium!
Das Elekterminal ist mit einem integrierten Baustein bestückt, der für den Datenaustausch mit dem Computer eine Schlüsselrolle spielt. Dieser Baustein trägt die Bezeichnung "UART", was so viel wie "Universeller asynchroner (Daten-)Sendeempfänger" bedeutet. Wie zwischen Terminal und Computer Daten gesendet und empfangen werden, das soll nun Gegenstand unserer Betrachtungen sein.

Vom Junior-Computer zur Peripherie und zurück

Serielle, bitweise Datenübertragung

Wir haben schon erwähnt, daß die große Tastatur und das Videodisplay plus TV-Empfänger (oder der Drucker) mit dem erweiterten Junior-Computer keine Einheit bilden, sondern selbständige Geräte sind. Solche peripheren Einheiten werden mit dem Computer über die RS-232-Schnittstelle verbunden. Bereits in Kapitel 10 wurde einiges über das RS-232-Interface gesagt, so daß wir jetzt darauf zurückgreifen können.
Das Drücken einer Taste ist eine menschliche Aktion, die der Computer im allgemeinen mit einer Rückmeldung auf dem Bildschirm beantwortet. Informationen fließen in zwei Richtungen: vom Elekterminal zum Computer und umgekehrt. Im Blockschema (Bild 1) sind deshalb zwei Leitungen

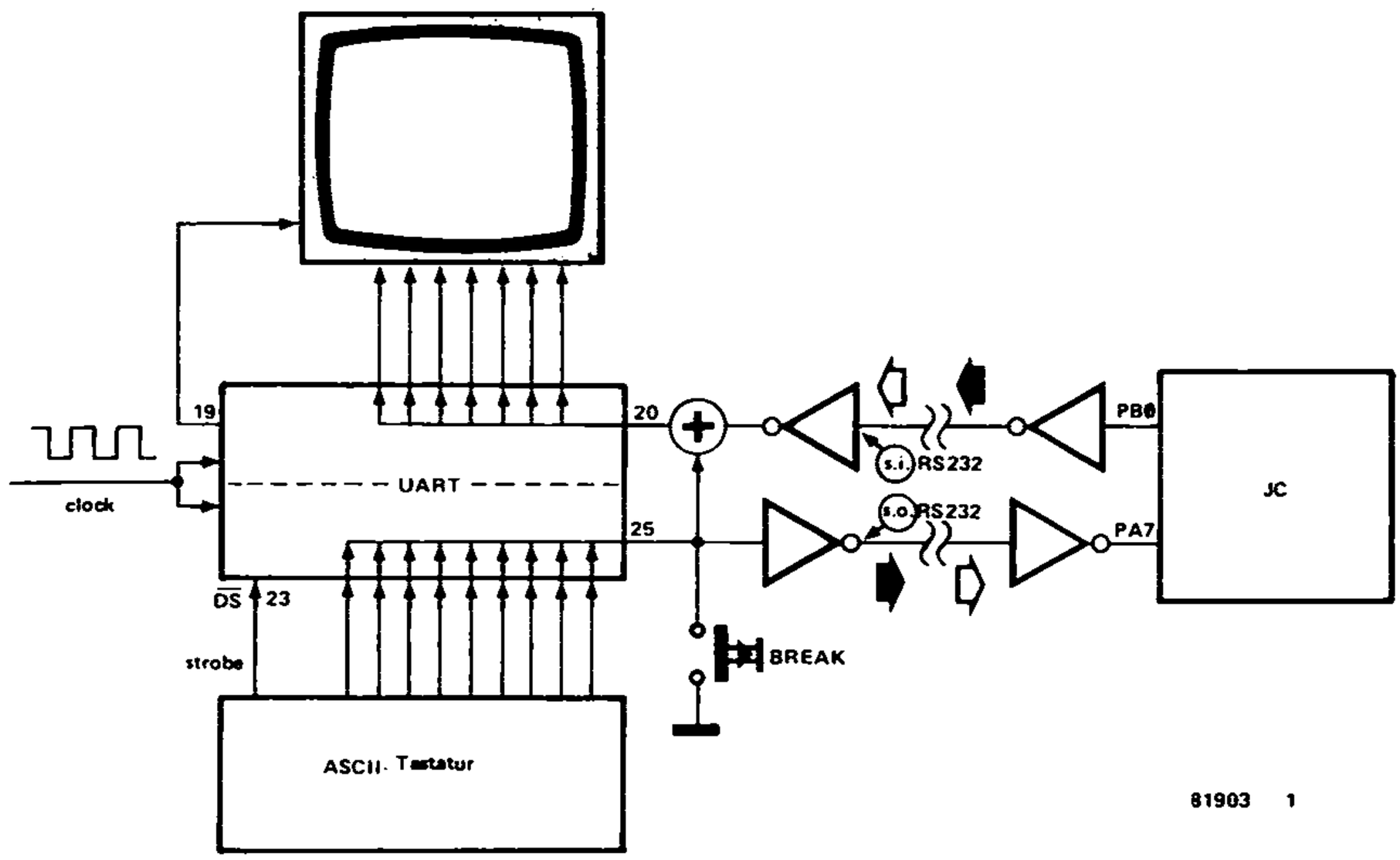


Bild 1. Blockschema des Datenaustauschs zwischen dem Junior-Computer und dem Elekterminal (Datensichtgerät mit ASCII-Tastatur).

mit der Bezeichnung "RS-232" vorhanden. Beide Leitungen verbinden jeweils einen Eingang und einen Ausgang. Der Eingang s.i. (serial in) des Elekterminals wird von dem als Ausgang geschalteten Port PBØ des Junior-Computers gesteuert, während der Ausgang s.o. (serial out) den als Eingang programmierten Port PA7 mit Informationen versorgt.
Im Elekterminal muß naturgemäß etwas mehr als nur zwei Leitungen stecken. Das zentrale Schaltzentrum ist dort der UART, über den im nächsten Abschnitt noch ausführlich berichtet wird. Die ASCII-Tastatur, die in Bild 1 links unten als Block dargestellt ist, steht mit dem UART über einen 8-Bit-Datenbus in Verbindung. Beim Drücken einer Taste wird der zugehörige ASCII-Kode – insgesamt 7 Bit – auf diesen Bus gesetzt. Das achte Bit ist für den Parity-Check reserviert; darüber später mehr. Das gilt auch für die Verbindung zwischen ASCII-Tastatur und UART im Zusammenhang mit dem Junior-Computer. Über die neunte Leitung (strobe) erfährt der UART, ob eine Taste gedrückt ist oder nicht. Das hier ankommende Signal ist das Startzeichen für die weitere Verarbeitung der von der Tastatur gelieferten Information.
Vom UART verläuft ein 7-Bit-Datenbus sowie eine Steuerleitung zu einem Block, in den ein Bildschirm gezeichnet wurde. Dieser Block symbolisiert die gesamte "Video-Fabrik" des Elekterminals. Über den 7-Bit-Datenbus gelangen die gerade erzeugten ASCII-Zeichen zum Bildschirm; sie stammen entweder von einer gedrückten Taste oder vom Junior-Computer. Wenn eine Taste gedrückt wird, lassen sich Buchstaben, Ziffern und Satzzeichen einerseits und grafische Kommandos wie CR (Wagenrücklauf), LF (nächste Zeile) usw. andererseits unterscheiden. Die Position auf dem Bildschirm, auf die das nächste Zeichen gesetzt wird, zeigt der sogenannte Cursor an. Dieser "Zeigestock" rückt nach jedem auf den

Bildschirm geschriebenen Zeichen um eine Stelle weiter; er läßt sich auch willkürlich verschieben, so daß jederzeit beliebig auf der Schirmfläche geschrieben werden kann.
Übrigens: Grafische Kommandos können nicht nur von der Tastatur, sondern auch vom Computer ausgehen. Zu diesem Zweck enthält das Systemprogramm Printer Monitor einige spezielle Subroutinen.
Dem UART wird außer den Datensignalen noch ein Taktsignal (clock) zugeführt, mit dessen Hilfe er die von der Tastatur kommenden parallelen Daten in serielle Daten und die vom Computer kommenden seriellen Daten in parallele Daten umwandelt. Der Vorgang ist mit dem vergleichbar, was beim Überqueren eines schmalen Flußübergangs durch eine Marschkolonne geschieht: Die Männer, die zunächst nebeneinander (parallel) marschiert sind, können den schmalen Steg nur nacheinander (seriell) passieren; anschließend bewegt sich die Kolonne in gewohnter Weise weiter. Der schmale Steg ist hier die Verbindung zwischen Junior-Computer und Terminal.
Die Wandlung und Rückwandlung des Datenformats wird von Bild 2 illustriert. Die ursprüngliche Situation ist nach Passieren der Engstelle bis auf einen kleinen Unterschied wiederhergestellt: Der Datenbus von der Tastatur zum UART besteht aus acht parallelen Leitungen, während der Bus vom UART zum Videoteil nur aus sieben Leitungen besteht.
In Bild 1 ist eine Querverbindung zwischen dem seriellen Ausgang Pin 25 des UART (dem Steg aus Bild 2) und dem seriellen Eingang Pin 20 des UART gezeichnet, die über einen Summationspunkt verläuft (die Abkürzungen s.i. und s.o. in Bild 1 betreffen den Eingang und Ausgang des Elekterminals, nicht des UART!). Der Kreis mit dem Pluszeichen soll andeuten, daß die für die Wiedergabe auf dem Bildschirm bestimmten Daten sowohl von der Tastatur als auch vom Computer stammen können. Allerdings dürfen Daten unterschiedlicher Herkunft nicht gleichzeitig eintreffen. Wenn die Querverbindung durchgeschaltet ist, wirkt sich ein Tastendruck

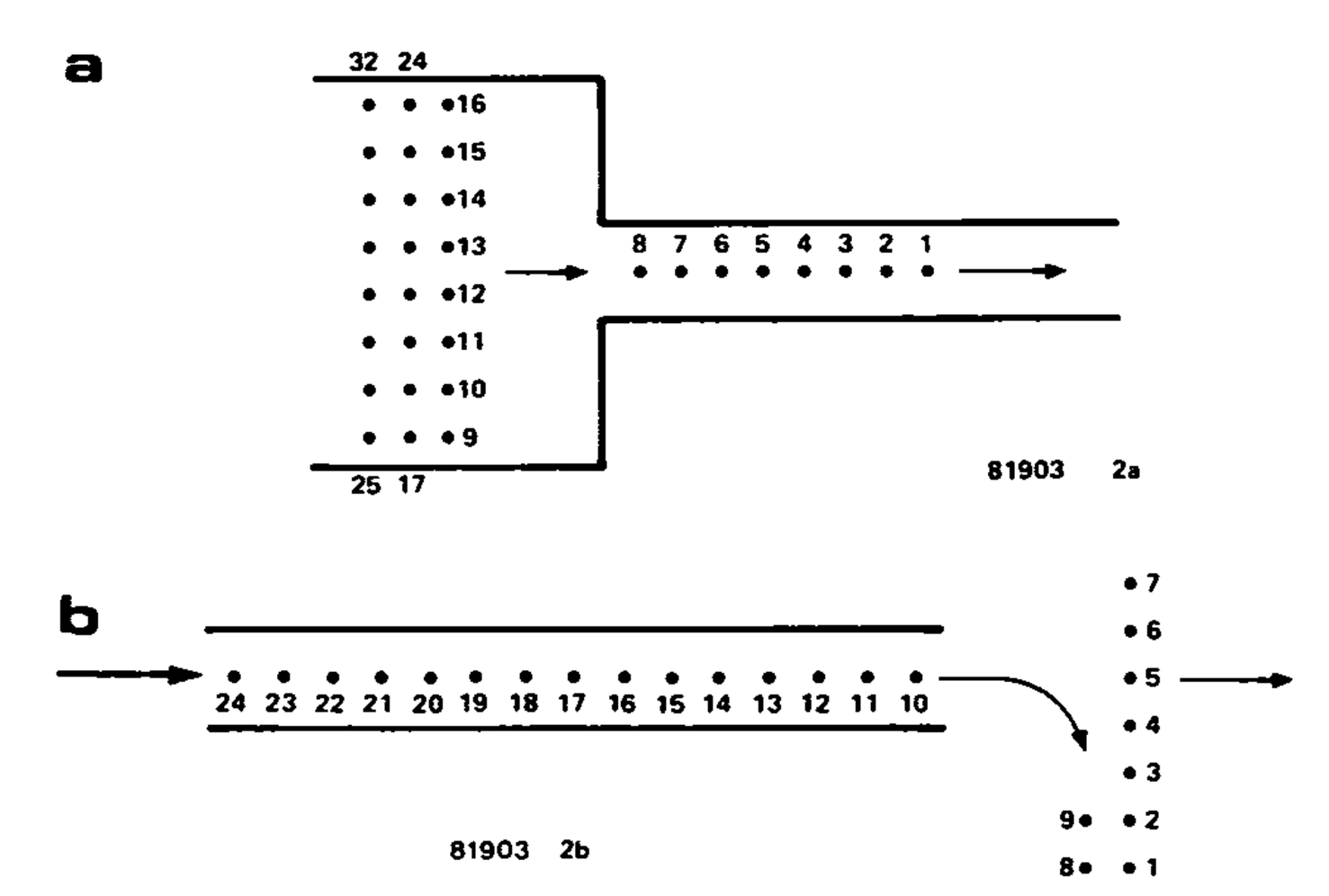


Bild 2. Das "Marschkolonnen-Modell" für die Umsetzung von parallelen in serielle Daten (a) und vom seriellen in das parallele Format (b).

unmittelbar auf das Geschehen aus, das der Bildschirm zeigt. Das ist in allen Fällen notwendig, in denen der Junior-Computer nach Drücken einer Taste keine Rückantwort an den UART sendet. Erzeugt also das Systemprogramm kein "Echo", so lassen sich die eingegebenen Daten trotzdem über die Querverbindung sichtbar machen.
In Bild 1 ist ferner eine Taste mit der Bezeichnung BRK zu sehen. Ein Druck auf diese Taste wirkt sich wie das Ziehen einer Notbremse aus. Wenn der Computer gerade mit dem Senden einer Information (Rückmeldung, Reaktion) beschäftigt ist, kann man ihm hiermit ins Wort fallen. Dies ist eine weitere Interrupt-Möglichkeit.
Übrig bleiben in Bild 1 noch die vier Inverter, deren Daseinsberechtigung auf den ersten Blick uneinsichtig ist. Zweifache Invertierung ergibt das gleiche Signal, so daß man die Inverter im Prinzip weglassen könnte. Die Norm RS-232 legt jedoch für die serielle Datenübertragung eine sogenannte negative Logik fest: logisch Ø ist identisch mit einer hohen und logisch 1 mit einer niedrigen Spannung. Da aber sowohl der Junior-Computer als auch das Elekterminal mit der vertrauten positiven Logik arbeiten, sind auf beiden Seiten der Leitungen Inverter notwendig. Man denke daran, daß an das RS-232-Interface des Computers auch andere periphere Geräte anschließbar sein müssen, die normgemäß mit negativer Logik arbeiten.

Gehende und kommende Bit

Die serielle Form, in der Daten zwischen Computer und Terminal ausgetauscht werden, ist eigentlich nur ein vorübergehender Zustand. Sowohl am Ursprungs- als auch am Bestimmungsort haben die Daten paralleles Format: Darauf deuten bereits die beiden Datenbusse in Bild 1 hin. Die vom Junior-Computer ausgehenden Daten sind ja die Inhalte von Speicherzellen (acht Bit parallel!), die ankommenden Daten werden wieder in Speicherplätze gesetzt. Der Computer wandelt das Datenformat (parallel nach seriell und umgekehrt) mit Hilfe von Software, während diese Aufgabe im Elekterminal von Hardware (den UART) erledigt wird.
Vergessen wir vorläufig den UART und betrachten wir statt dessen, wie eine Tasteninformation in serieller Form zum Junior-Computer und über die Querverbindung zum Bildschirm gelangt. Bild 3a gibt hierüber Aufschluß. Dargestellt sind die Signale vor der Invertierung auf seiten des Elekterminals und nach der Invertierung auf seiten des Junior-Computers. Im Ruhezustand ist das gesendete Signal logisch 1. Wenn nun ein ASCII-Zeichen übertragen werden soll, muß der Gegenseite zuerst der Beginn der Sendung signalisiert werden. Das geschieht dadurch, daß das Signal kurzzeitig auf logisch Ø gesetzt wird. Diese Zeit ist die sogenannte Bitzeit, die logische Ø ist das
Startbit.
Der Computer reagiert auf das Startbit wie folgt: Er wartet 1,5 Bitzeiten nach dem ersten Spannungswechsel ab und prüft dann, ob das ankommende Signal logisch Ø oder logisch 1 ist. Das Ergebnis hält er in einem Register fest, denn dies ist das Bit bØ des ASCII-Zeichens. Danach wartet der Computer eine weitere Bitzeit, prüft wieder das ankommende Signal, setzt dieses neben das erste in das Register usw. Auf diese Weise stehen

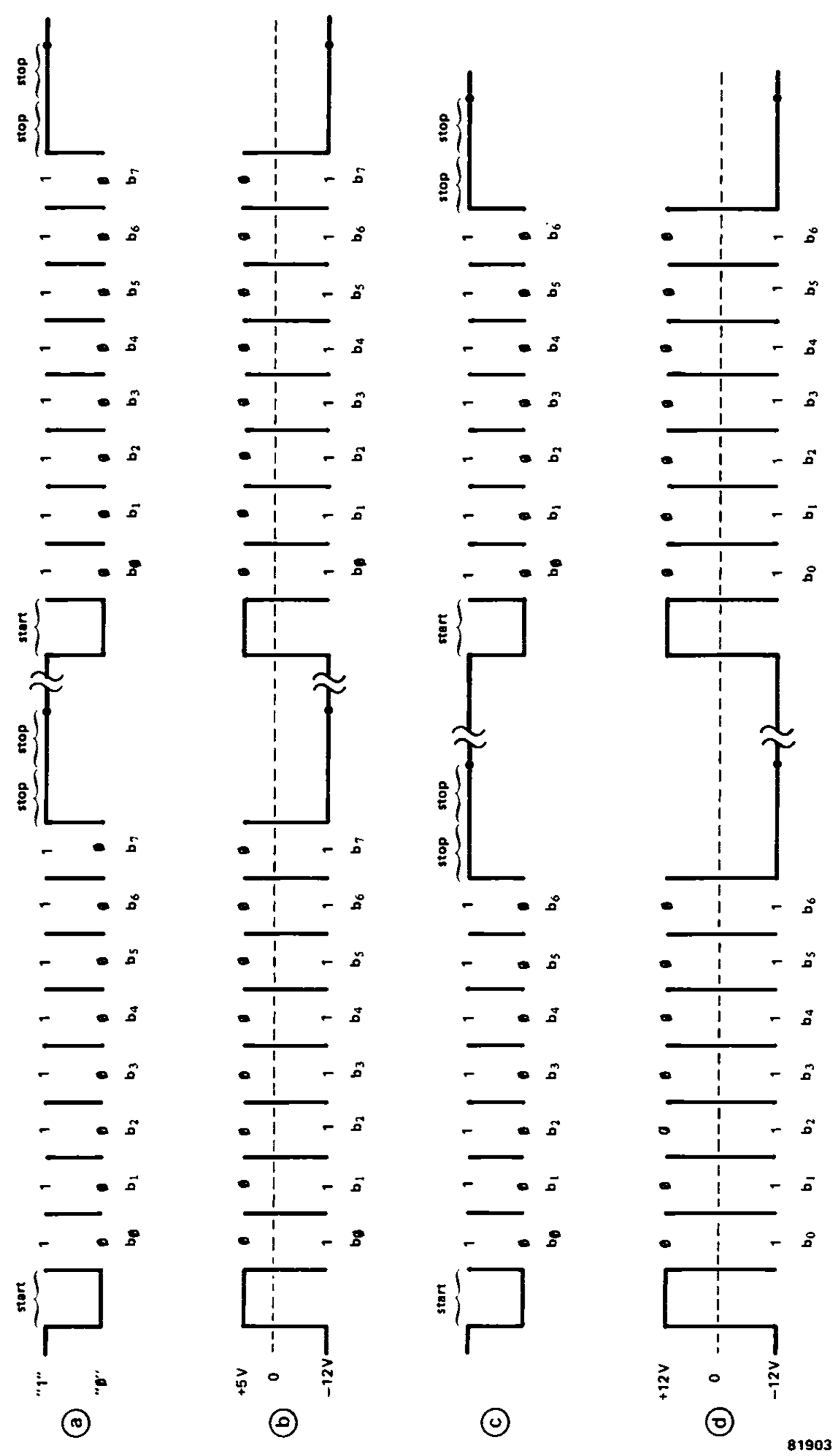


Bild 3. Die logischen Signale (a und c) und die Spannungen (b und d) beim Senden von zwei aufeinander folgenden ASCII-Zeichen in Richtung von der Tastatur über den UART zum Computer (a und b) und vom Computer über den UART zum Bildschirm (c und d).

schließlich die Bit bØ . . . b7 parallel in dem Register. Bit bØ ist das Bit des 7-Bit-ASCII-Kodes, das ganz rechts steht, Bit b6 steht ganz links. Das achte Bit (b7) ist das sogenannte
Paritätsbit.
Dieses Bit kann zur Kontrolle auf richtigen Empfang des ASCII-Kodes herangezogen werden. Obwohl hiervon beim Junior-Computer kein Gebrauch gemacht wird (vergleiche Bild 3c), wollen wir in diesem Zusammenhang die Rolle des Paritätsbit kurz streifen. Man kann dem Paritätsbit b7 (Junior-Format) einen solchen Wert (Ø oder 1) geben, daß die Anzahl der im Datenwort b6 . . . bØ vorkommenden logischen Einsen geradzahlig oder auch ungeradzahlig ist. Im ersten Fall spricht man von gerader, im zweiten von ungerader Parität. Auf der Empfangsseite kann dann geprüft werden, ob noch die richtige Parität vorhanden ist. Wenn sie nicht mehr stimmt, dann löst dies eine Fehleranzeige oder eine andere Maßnahme des Computers aus. Das kann zum Beispiel eine Aufforderung an die Sendeseite sein, das falsch empfangene ASCII-Zeichen noch einmal zu wiederholen. Die beschriebene Kontrolle ist allerdings bei weitem nicht absolut zuverlässig.
Nachdem die acht Bit auf der Empfängerseite angekommen sind, muß die Leitung wieder ihren Ruhezustand einnehmen. Deshalb folgt noch mindestens ein *Stopbit;* in unserem Fall sind es zwei.
In Bild 3b sind die Signale auf der Leitung dargestellt. Der einzige Unterschied zu Bild 3a ist die Invertierung, die die RS-232-Norm verlangt. Diode D4 im Elekterminal ist bei den gezeichneten Signalen nicht vorhanden.
Wie werden die Daten in umgekehrter Richtung, vom Junior-Computer zum Elekterminal transportiert? Darüber geben die Bilder 3c und 3d Auskunft: Ein Paritätsbit wird nicht übertragen, und die hohe Spannung auf der RS-232-Leitung beträgt +12 V statt +5 V.
Fast hätten wir's vergessen: Das Bit b7 wird zwar wie bei der Kontrolle auf gerade Parität von der Tastatur über den UART zum Junior-Computer geschickt; dieser macht es jedoch nach dem Empfang zu Null.

Der UART

Die Abkürzung "UART" ist inzwischen mehrfach aufgetaucht; wir wollen nun sehen, was sich dahinter verbirgt. UART bedeutet "Universal Asynchronous Receiver/Transmitter", oder übersetzt "Universeller, asynchroner (Daten-)Sendeempfänger". Dieser zum Elekterminal gehörende integrierte Baustein sorgt sowohl für die Umsetzung von seriellen in parallele Daten (vom Junior-Computer zum Bildschirm) als auch für die Wandlung von parallelen in serielle Daten (von der Tastatur zum Junior-Computer und zum Bildschirm). Asynchron arbeitet der UART deshalb, weil die Wandelgeschwindigkeit von den Bitzeiten und nicht von einem getrennt übermittelten Taktsignal abhängt.
Im Artikel über das Elekterminal wurde bereits das eine und andere zum UART gesagt. Als Ergänzung zeigt Bild 4 die vollständige Anschlußbelegung des UART-ICs. In Bild 5 ist derjenige Teil der Elekterminal-Schaltung wiedergegeben, der den UART und seine nähere Umgebung betrifft. Hier wurden gegenüber der Originalschaltung einige kleinere Änderungen

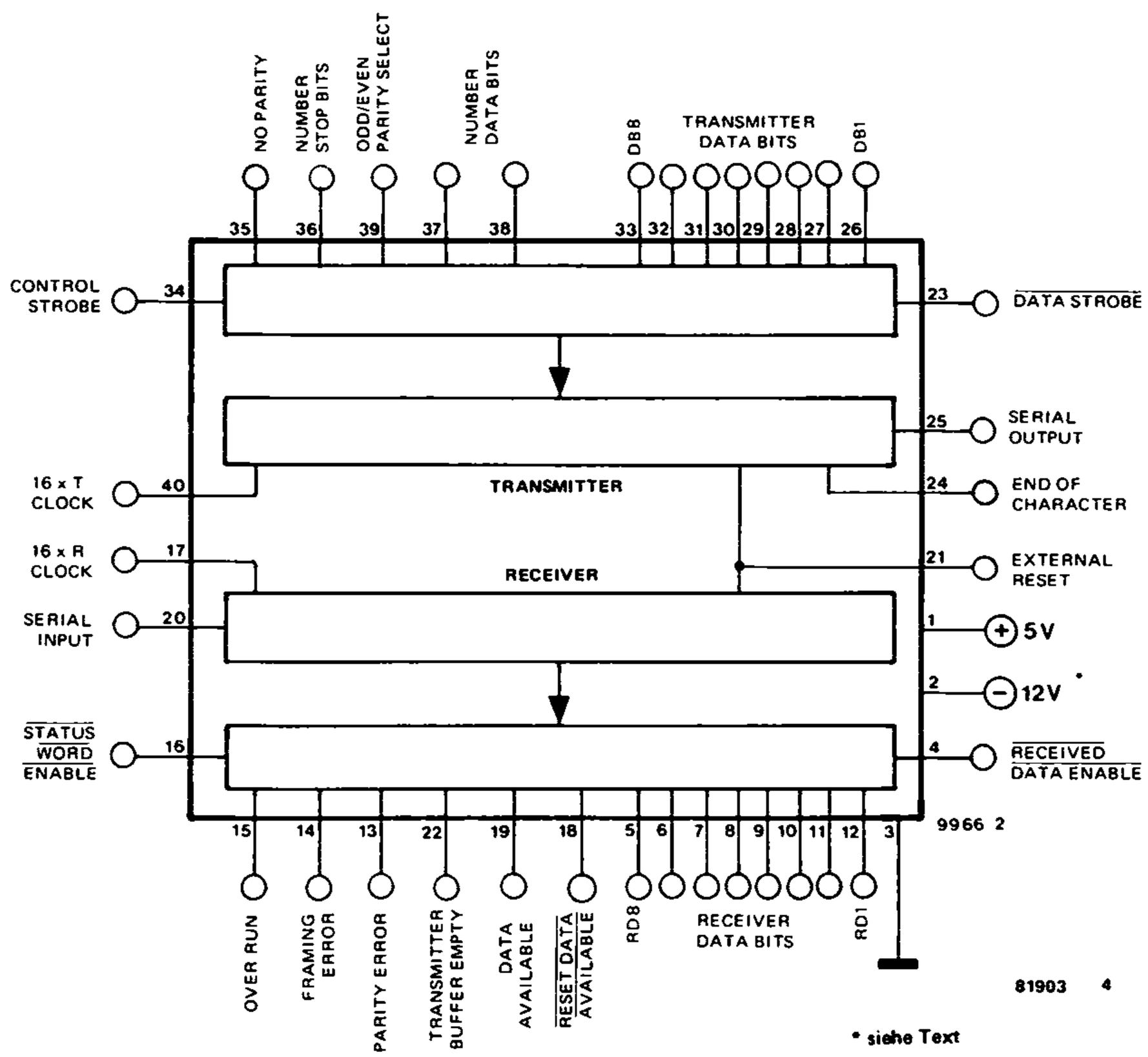


Bild 4. Anschlußbelegung des im Elekterminal verwendeten UART (Universal Asynchronous Receiver/Transmitter).

vorgenommen; sie werden im Abschnitt über den Bau des Elekterminals noch ausführlich besprochen. Bild 6 macht am Modell des Fließbandes das Prinzip der seriellen Datenübertragung und gleichzeitig die Arbeitsweise des UART deutlich. Dieses Bild wollen wir nun betrachten.
Man stelle sich zwei Fließbänder vor: Ein Band läuft vom Junior-Computer zum Elekterminal, das andere in entgegengesetzter Richtung. Auf beiden Fließbändern liegen hintereinander angeordnet schwarze und weiße Kugeln. Eine weiße Kugel ist gleichbedeutend mit einem Bit, das logisch 1 ist, eine schwarze Kugel symbolisiert eine logische Ø, und eine Kugel mit einem X ist eine "Don't-Care-Kugel"; sie kann entweder schwarz oder weiß sein. Ist das Band um die Länge eines Kugeldurchmessers weitergelaufen, dann ist eine Bitzeit verstrichen. Die Bandgeschwindigkeit steigt also mit abnehmender Bitzeit, oder anders ausgedrückt, mit zunehmender Baudrate an.
Bild 6 gibt die Verhältnisse vor und hinter den RS-232-Leitungen wieder. Auf den Leitungen sind wegen der Normvorschrift RS-232 die schwarzen

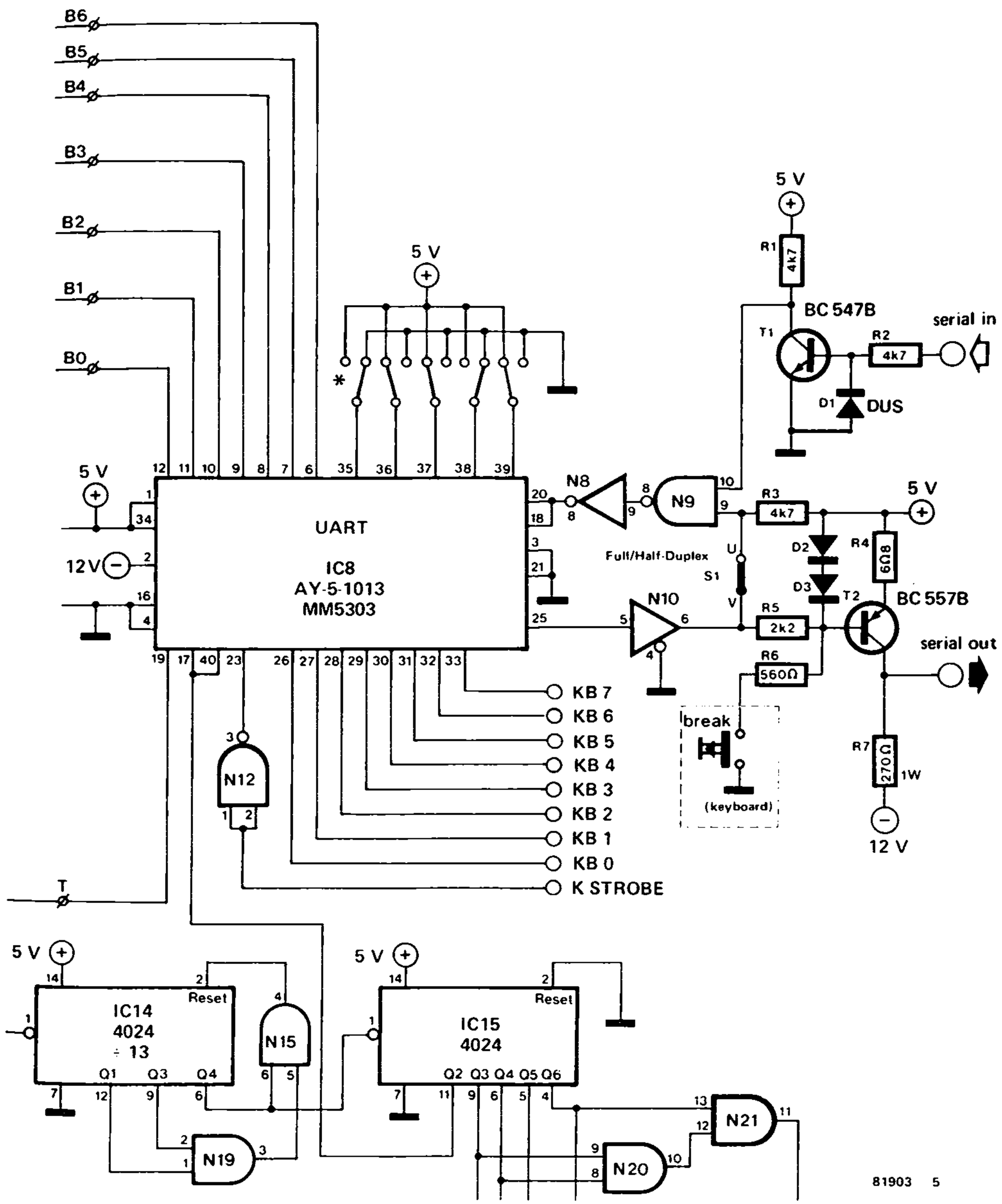


Bild 5. Teilschaltung des Elekterminal mit dem UART (IC8). Um das Elekterminal an den Junior-Computer anschließen zu können, sind einige kleinere Modifikationen notwendig.

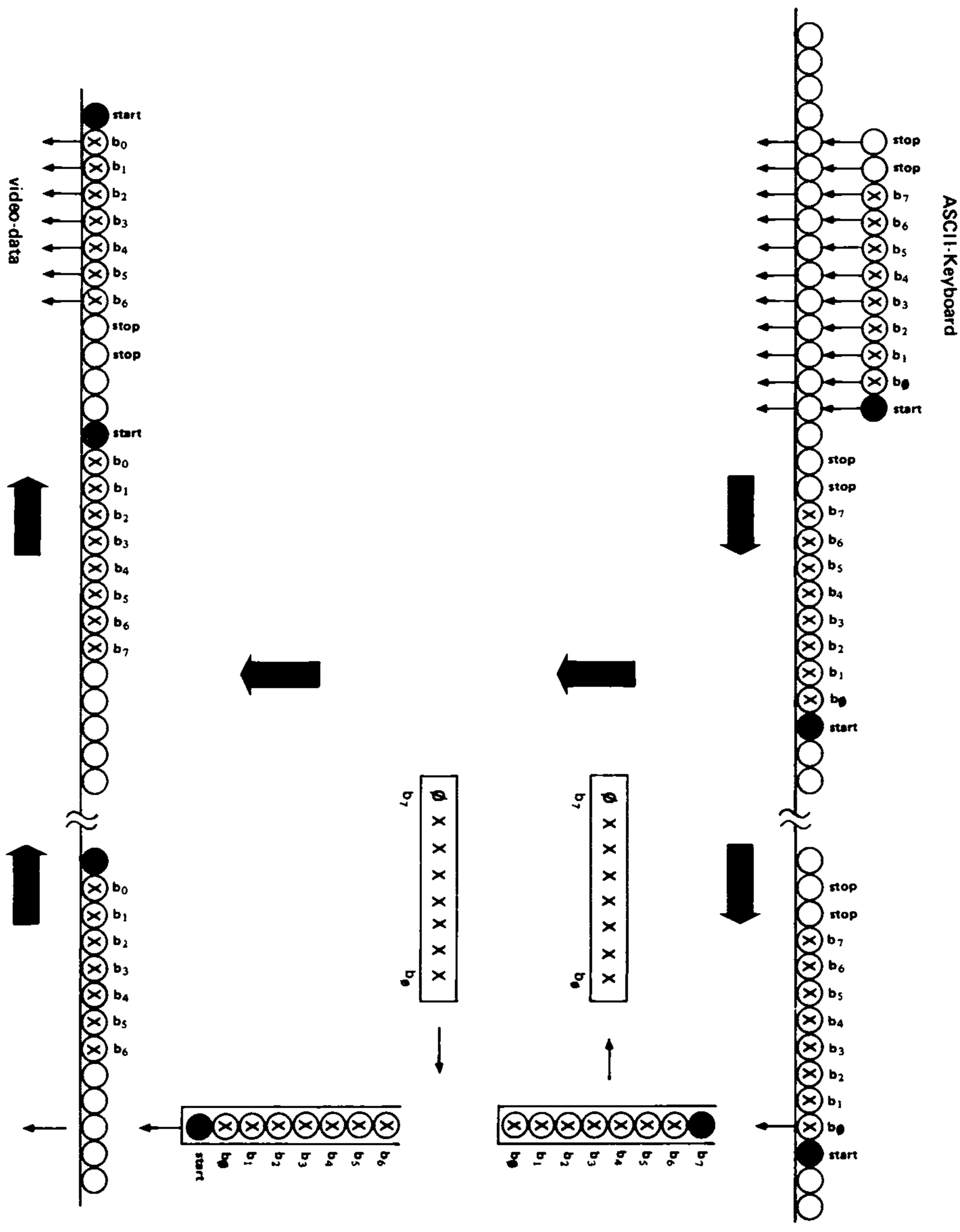


Bild 6. Das "Fließband-Modell" für den seriellen Datentransport einschließlich der Umsetzung vom parallelen in das serielle und vom seriellen in das parallele Datenformat.

Kugeln weiß und die weißen Kugeln schwarz; dies ändert jedoch nichts am Prinzip.
Beginnen wir mit dem Drücken einer Taste auf der ASCII-Tastatur. Wir stellen uns elf Röhren vor, die an einem bestimmten Ort (links oben in Bild 6) oberhalb des Fließbandes aufgestellt sind. Beim Drücken der Taste befinden sich unter den mit einer Klappe verschlossenen Röhren nur weiße Kugeln auf dem Fließband: Der Versand des vorgegangenen ASCII-Zeichens ist abgeschlossen. Die Röhrenverschlüsse werden gleichzeitig geöffnet; die Kugeln fallen aus den Röhren und nehmen den Platz der Kugeln auf dem Fließband ein. Letztere verschwinden, und außerdem läuft der gesamte Vorgang unendlich schnell ab (es handelt sich hier nur um ein Modell!).
Die schwarze Kugel, die das Startbit verkörpert, befindet sich rechts von den nachfolgenden Datenbits bØ . . . b7. Das stimmt mit der Laufrichtung des Bandes vom Elekterminal zum Junior-Computer überein. Die Datenbit-Kugeln sind in der Reihenfolge angeordnet, die sie auch innerhalb des Byte einnehmen, zu dem sie gehören: bØ steht ganz rechts und b6 (abgesehen von Paritätsbit b7) ganz links. Die beiden folgenden weißern Kugeln sind die Stopbits; sie bilden das Schlußlicht des Datentransports.
Das obere Fließband in Bild 6 bewegt sich in Richtung zum Junior-Computer. Dort sehen wir eine vertikal angeordnete Röhre, die acht Kugeln aufnehmen kann. Sobald eine schwarze Kugel (das Startbit) erscheint, fallen die folgenden acht Kugeln der Reihe nach in diese Röhre. Die letzte ankommende Kugel (das Paritätsbit b7), die in die Röhre fällt, färben wir schwarz. Wie schon an anderer Stelle erwähnt, sind wir an der Paritätskontrolle nicht interessiert. In der folgenden Verarbeitungsphase setzt der Junior-Computer mittels Software die eingetroffenen Kugeln zu einem Byte (Inhalt einer Speicherzelle) zusammen. Die Röhre schütten wir sozusagen nach links hin aus.
Das untere Fließband in Bild 6 läuft vom Junior-Computer zum Elekterminal. Hier beginnt das Versenden von Daten beim Computer. Der Inhalt einer Speicherzelle wird in eine oberhalb des Bandes vertikal angeordnete Röhre gesetzt, die acht Kugeln aufnehmen kann. Wenn sich auf dem Band unterhalb der Röhre nur weiße Kugeln befinden (das vorangegangene ASCII-Zeichen einschließlich der beiden "weißen" Stopbits ist vollständig abgesandt), öffnet sich der Verschluß am unteren Ende der Röhre, so daß eine Kugel nach der anderen auf das Band fällt. Die dort vorher liegenden weißen Kugeln werden von ihrem Platz verdrängt. Das Schlußlicht bilden wieder die beiden weißen Kugeln, die die beiden Stopbits repräsentieren. Die bedeutet nichts anderes, als daß das Senden des nächsten ASCII-Zeichens für die Dauer von zwei Bitzeiten blockiert wird. Auf der Empfangsseite geschieht genau das Gleiche: Unmittelbar nach Ankunft des letzten Datenbit (b6 oder b7) ist der Empfänger für die Dauer von zwei Bitzeiten gesperrt.
Zurück zum Fließbandmodell in Bild 6: Trifft beim UART im Elekterminal ein Startbit ein (dies ist die erste schwarze Kugel nach mindestens zwei aufeinander folgenden weißen Kugeln), dann nimmt der UART die auf die schwarze Kugel folgenden sieben Kugeln gleichzeitig vom Band. Diese sieben Bit sind der ASCII-Kode des Zeichens oder grafischen Kommandos, das auf den Bildschirm geschrieben bzw. ausgeführt werden

soll.
Noch eine Anmerkung zu Bild 6: Die von der Tastatur stammenden Informationen werden nicht nur zum Junior-Computer transportiert, sondern gelangen auch auf internen Wegen zum Bildschirm. In dieser Hinsicht ist Bild 6 unvollständig. Zwischen den von der Tastatur und den vom Computer kommenden Informationen (der Antwort) gibt es trotzdem keinen "Zusammenstoß", weil der Computer nicht unendlich schnell arbeiten kann. Bis seine Reaktion eintrifft, ist das Zeichen längst auf den Bildschirm geschrieben oder das grafische Kommando ausgeführt.

Die Praxis

Elekterminal mit ASCII-Tastatur: Bau und Anschluß an den Junior-Computer

Bereits an anderer Stelle wurde in diesem Kapitel darauf hingewiesen, daß das Elekterminal und die zugehörige ASCII-Tastatur frühere, in der Zeitschrift "Elektor" veröffentlichte Bauprojekte sind, und daß auch das Buch "µComputer für den Selbstbau, Teil 2" die Originalartikel enthält. Dort sind die vollständigen Schaltungen, die Schaltungsbeschreibungen, die Platinenlayouts sowie Hinweise zum Bau und zur Inbetriebnahme zu

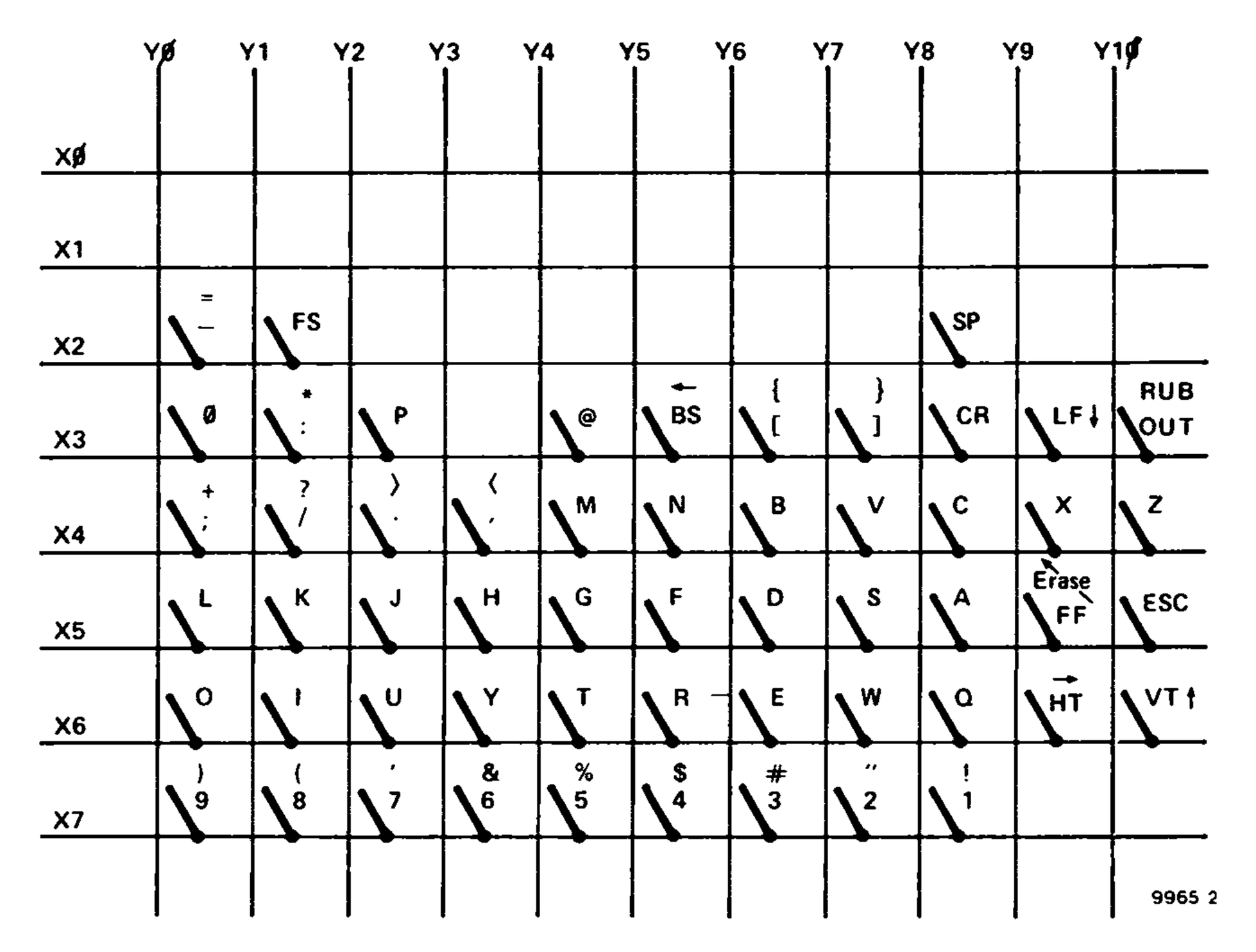


Bild 7. Die Tastenmatrix der ASCII-Tastatur. Reservetaste RES erhält die Funktion RUB OUT.

finden. Leider ist es nicht möglich, dies alles im Rahmen der Junior-Computer-Buchreihe noch einmal abzudrucken, obwohl es den Autoren durchaus wünschenswert erscheint. Die Anzahl der Seiten ist jedoch aus naheliegenden Gründen begrenzt, so daß ein anderes Kapitel zu kurz kommen würde.
Eine Schritt-für-Schritt-Bauanleitung für das Elekterminal erscheint in diesem Stadium des Junior-Computer-Ausbaus auch nicht mehr notwendig. Die erforderlichen Kenntnisse und Erfahrungen sind jetzt vorhanden: Es entstanden inzwischen die Basisversion des Juniors, die erweiterte Speisung und die Interfacekarte, so daß der Bau des Elekterminals mit der ASCII-Tastatur und (eventuell) dem UHF/VHF-Modulator fast nur noch ein Kinderspiel ist. Außerdem sind die Platinen aller weiteren Bauaktivitäten nicht mehr doppel-, sondern nur noch einseitig mit Leiterbahnen bedruckt. Nachfolgend geben wir deshalb lediglich Hinweise für den Anschluß des Elekterminals an den Junior-Computer. Wir glauben, daß auch so keine unlösbaren Probleme auftauchen werden.

Zuerst die Tastatur . . .

Beim Bau der ASCII-Tastatur sind in Zusammenhang mit dem Junior-Computer folgende Punkte zu beachten:

- Die Taste RES (Reserve) ist ursprünglich nicht an die Tastenmatrix X0 . . . X7/Y0 . . . Y10 angeschlossen. Dies muß nun geschehen: Aus Bild 7 geht hervor, daß ein Tastenkontakt mit der Matrixzeile X3 und der andere Tastenkontakt mit der Matrixspalte Y10 zu verbinden ist. Taste RES erhält dadurch die Funktion DELETE oder RUB OUT. Das Drücken dieser Taste löst den eigentlichen Start des Systemprogramms PM aus. Bild 8 zeigt einen Teil der Tastaturplatine mit den beiden genannten Verbindungen. Matrixzeile X3 liegt an Pin 36 von IC1 (AY-5-2376), während Matrixspalte Y10 mit Pin 21 des gleichen ICs verbunden ist.
- Die Verbindungen b und c werden hergestellt; ein Kontakt der BRK-Taste wird an Masse gelegt (siehe Bild 9).
- Der Aufdruck auf der Tastaturplatine kennzeichnet nicht die einzelnen Ausgänge des ICs. Die Bezeichnungen sowie die entsprechenden Anschlüsse auf der Videointerface-Platine können jedoch Bild 9 entnommen werden. Zu beachten ist, daß nicht S6, sondern S8 mit Anschlußpunkt KB5 des Elekterminals verbunden wird. Dadurch erzeugt die Tastatur beim Drücken einer Buchstabentaste automatisch den ASCII-Kode des betreffenden Großbuchstaben; eine der beiden SFT-Tasten braucht nicht gedrückt zu werden. Der Zeichengenerator des Videointerface setzt nämlich nur große Buchstaben auf den Bildschirm. Ferner darf die −12 V-Leitung nicht fehlen!

. . . und dann das Videointerface

Bild 5 zeigt einen Ausschnitt aus der Schaltung des Videointerface, der alle für den Anschluß an den Junior-Computer wichtigen Einzelheiten enthält. In Bild 10 ist der verkleinerte (!) Bestückungsaufdruck der Videointerface-Platine mit verschiedenen Verdrahtungsanweisungen wiedergegeben. Um das Videointerface an den Junior-Computer anzupassen, ist folgendes notwendig:

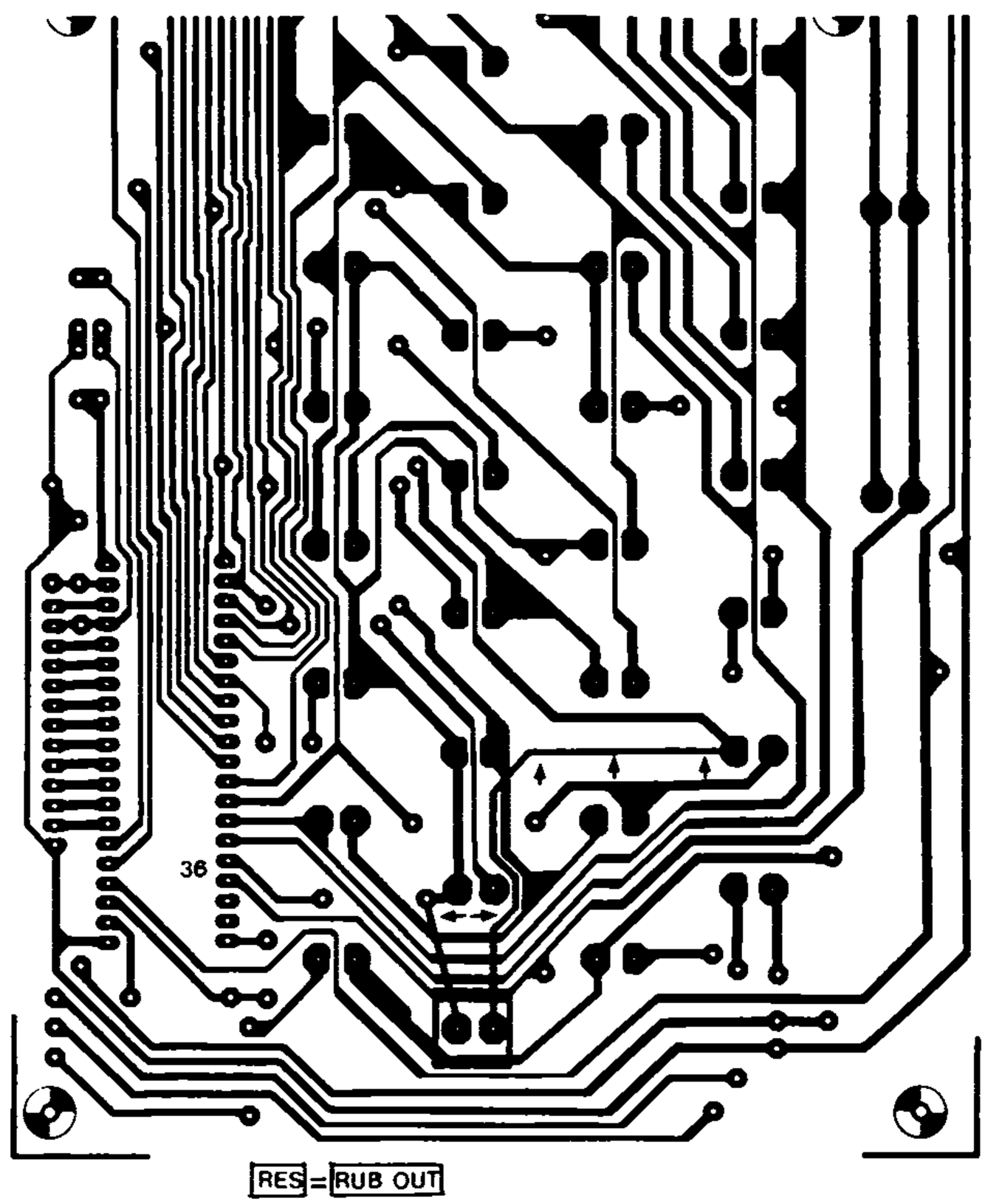


Bild 8. So wird die RES-Taste angeschlossen, damit sie die Funktion RUB OUT erhält.

- Zuerst Diode D4. Diese Diode entfällt; sie fehlt deshalb bereits in Bild 10.
- Der Baudrateschalter S2a/S2b entfällt ebenfalls, denn der Junior-Computer arbeitet mit der festen Baudrate 1200 Bd. Der Mittenkontakt MS2b wird mit Masse verbunden, der Kontakt MS2a wird an "1200" gelegt (siehe Bild 10).
- Schalter S1 (Full/Half Duplex) kann durch eine Drahtbrücke zwischen den Punkten U und V ersetzt werden. Bleibt der Schalter an seiner Stelle, so muß er stets geschlossen sein. Nur dann entsteht ein Hardware-Echo der gedrückten Taste auf dem Bildschirm.
- Die Brücken an den Punkten 35 . . . 39 werden entsprechend dem für den Junior gewählten Datenformat gelegt: Paritätsbit vorhanden, gerade Parität, zwei Stopbit und sieben Bit pro ASCII-Zeichen (gebräuchliche Form des ASCII-Kodes). Zwar nimmt das Systemprogramm PM keine Paritätskontrolle vor, das Paritätsbit muß jedoch aus anderen Gründen mitgesendet werden. Um die Lage der Brücken 36 und 37 (nach +5 V) deutlicher darstellen zu können, fehlen in Bild 10 die dazwischenliegenden Anschlüsse.

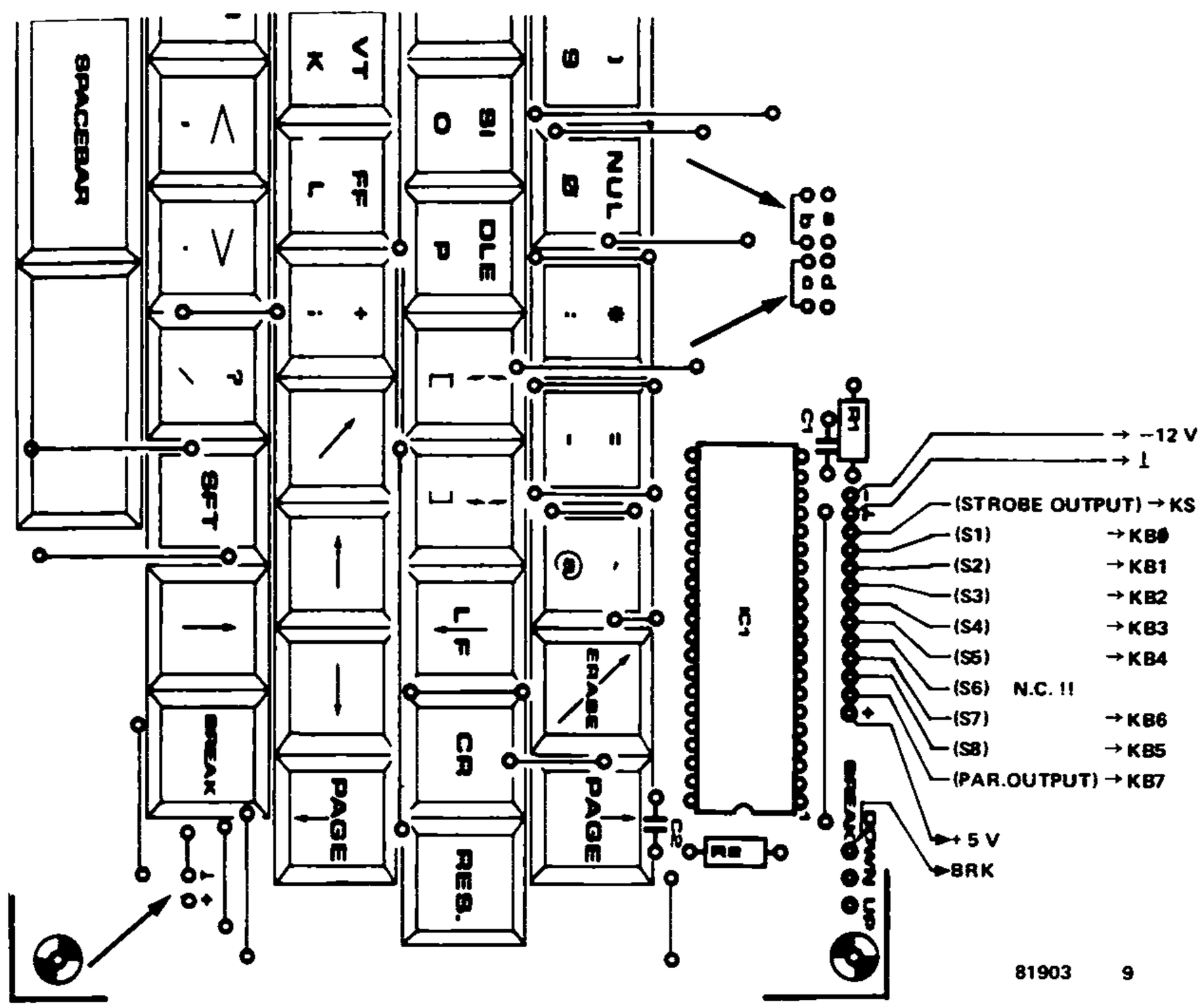
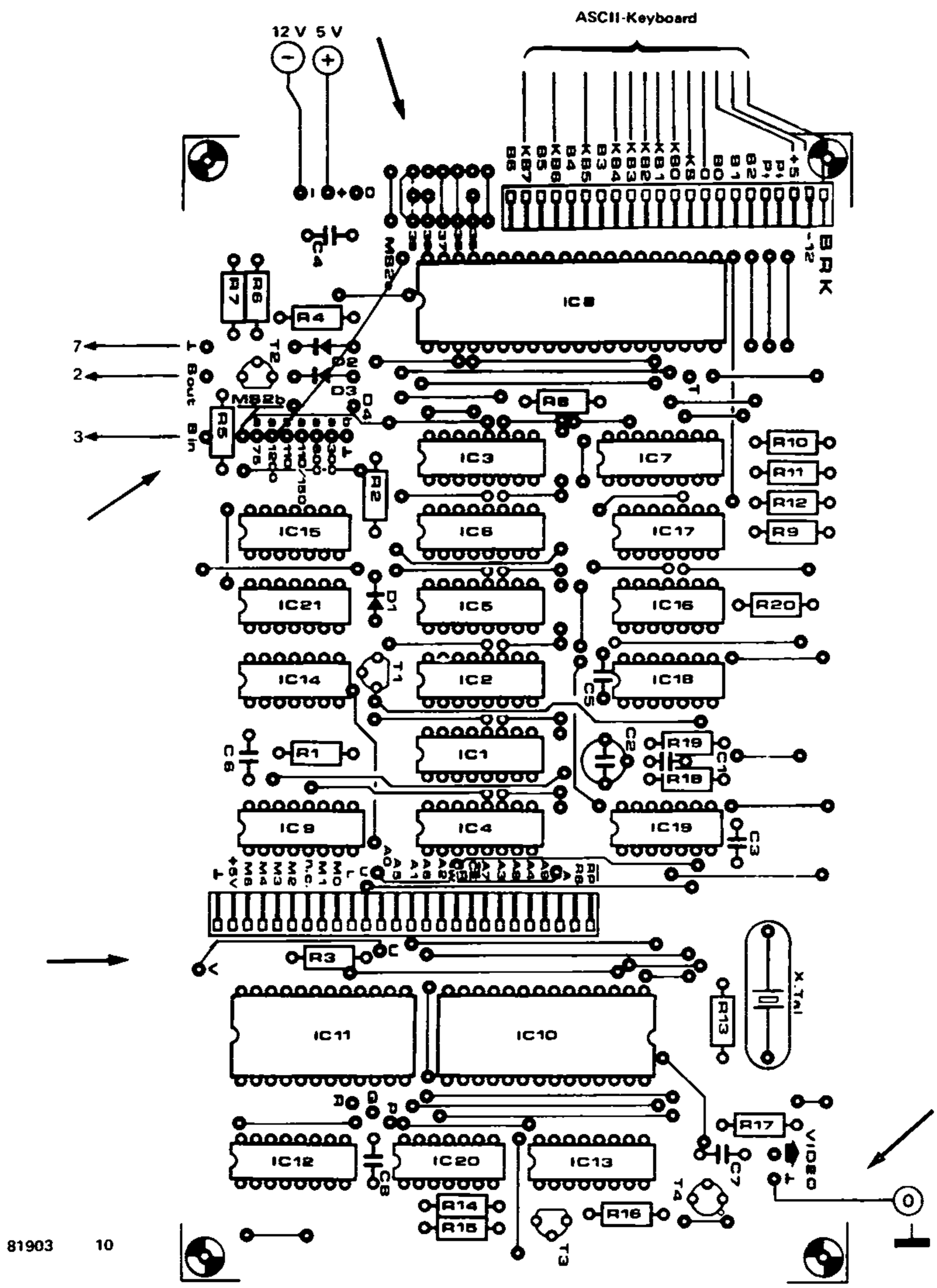


Bild 9. Die Drahtbrücken und Anschlüsse der ASCII-Tastatur.

- Ohne Seitenerweiterung des Elekterminals ist das Videointerface, wie Bild 10 zeigt, über 13 Leitungen mit der Tastatur verbunden. Für den Anschluß auf der Elekterminal-Platine kann ein geeigneter Konnektor verwendet werden.
- Als **UART** (IC8) verwende man möglichst einen Typ, der mit nur einer Speisespannung (+5 V) auskommt. Der Strombedarf im −12 V-Zweig des Netzteils ist dann minimal; das Elekterminal kann aus dem erweiterten Junior-Netzteil gespeist werden. Insgesamt nimmt das Elekterminal einschließlich der Tastatur etwa 500 mA an Strom auf. Im Handel sind verschiedene UART-Typen erhältlich, die nur die Speisespannung +5 V benötigen: AY-3-1015 (General Instrument), HM6402 (Harris), COM8017 und COM8502 (SMC).
- Die Netzteilanschlüsse Masse, +5 V und −12 V sind ebenfalls in Bild 10 angegeben. Man legt die zentrale Masseleitung am besten an den Masseanschluß des Video-Ausgangs.
- Die Verbindungen zwischen dem Video-Ausgang und dem TV-Gerät oder dem UHF/VHF-Modulator müssen über Koaxialkabel mit einem Wellenwiderstand von 50 . . . 75 Ω hergestellt werden.
- **Seitenerweiterung:** Im Buch "µComputer für den Selbstbau, Teil 2" ist beschrieben, wie man den Textspeicher des Videointerface von 1 K (16 Zeilen) auf 4 K 64 Zeilen) erweitern kann. Gesteuert werden die Sei-

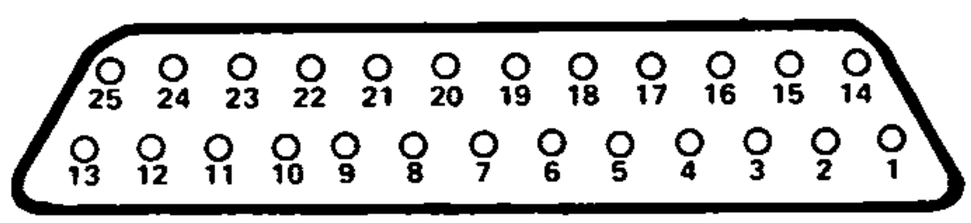


81903 10

Bild 10. Die Drahtbrücken und Anschlüsse auf der Elekterminal-Platine. Das Platinenlayout mußte hier verkleinert wiedergegeben werden.

ten mit den beiden Hardwaretasten PAGE ↑ und PAGE ↓ der ASCII-Tastatur, so daß sich die Anzahl der Leitungen zwischen Videointerface und Tastatur um zwei erhöht. Der Anschluß "DOWN" der Tastatur wird mit Punkt "P↓" der Videointerface-Platine verbunden, Anschluß "UP" wird an Punkt "P↑" gelegt. Siehe hierzu Bild 9 und Bild 10.

- **Anschluß an den Junior-Computer.** Welche Verbindungen sind notwendig? Zu allererst natürlich die beiden RS-232-Leitungen für den seriellen Datentransport. Ferner mindestens eine oder besser zwei Masseleitungen: einmal die "digitale" Masse und dann die Masseleitung zum Netzteil. Es fehlen noch die Speisespannungsleitungen für +5 V und −12 V. Insgesamt sind fünf bzw. sechs Leitungen zu verlegen, so daß bei nur einer Masseleitung eine fünfpolige DIN-NF-Steckverbindung verwendbar ist. Natürlich kann das Elekterminal auch fest mit dem Computer verkabelt werden. Auf jeden Fall sind aber die richtigen Punkte miteinander zu verbinden: "Serial Out" in der Videointerface-Schaltung ist gleichbedeutend mit "Sout" auf der Platine; das Gleiche gilt für "Serial In" und "Sin". Wohin führen die Leitungen, die hier angelötet sind? Schlagen wir noch einmal Bild 2 aus Kapitel 10 auf. Der Punkt "Sout" der Videointerface-Platine muß mit Pin 2 des D-Konnektors auf der Interfacekarte verbunden sein; Punkt "Sin" muß an Pin 3 des D-Konnektors liegen.
Eine professionelle Lösung für das Verbinden von Zentraleinheit (Junior-Computer) und Peripherie (Elekterminal) sind 25-polige D-Konnektoren (weiblich) an beiden Einheiten, die über ein steckbares Kabel mit 25-poligen D-Konnektoren (männlich) auf beiden Seiten in Kontakt stehen. In Bild 11 ist die Kontaktanordnung eines männlichen 25-poligen D-Konnektors gezeichnet; darunter ist die Belegung der einzelnen Kontakte angegeben, soweit sie allgemein gebräuchlich ist. Aus diesem Bild wird gleichzeitig deutlich, daß die Norm EIA-RS232C ihren Ursprung im angloamerikanischen Sprachraum hat. Für die Verbindungen zwischen Junior-Computer und Elekterminal spielen ausschließlich die Kontakte 2, 3 und 7 eine Rolle. Bei den Kontakten 2 und 3 ist erhöhte Aufmerksamkeit geboten: Verwirrung entsteht durch die Begriffe "gesendete Daten"



1. Protective Ground
2. Transmitted Data
3. Received Data
4. Request to Send
5. Clear to Send
6. Data Set Ready
7. Signal Ground
8. Data Carrier Detect
9 . . . 14 Not Used
15. Transmitted Bit Clock Internal
16. Not Used
17. Received Bit Clock
18,19. Not Used
20. Data Terminal Ready
21. Not Used
22. Ring Indicator
23. Data Signal Rate Selector
24. Transmitted Bit Clock
25. Not Used

81903 11

Bild 11. Anschlüsse des 25-poligen D-Konnektors mit ihren Funktionen.

(Transmitted Data) und "empfangene Daten" (Received Data). Was nämlich von der einen Seite gesendet wird, empfängt die andere Seite, und umgekehrt. Betrachten wir die Anschlüsse des D-Konnektors auf der Interfacekarte (Bild 2 in Kapitel 10), dann stellen wir fest, daß über Kontakt 2 Daten vom Computer empfangen und über Kontakt 3 gesendet werden. Das steht auf den ersten Blick nicht im Einklang mit Bild 11, wo Kontakt 2 für "Transmitted Data" und Kontakt 3 für "Received Data" bestimmt ist. Die Begriffe "gesendet" (transmitted) und "empfangen" (received) beziehen sich hier jedoch auf die Peripherie und nicht auf die Zentraleinheit. Vom Junior-Computer aus gesehen müßten die Bezeichnungen gegeneinander vertauscht werden. Wie dem auch sei, auf jeden Fall ist Punkt 2 des RS-232-Anschlusses auf der Interfacekarte des Junior-Computers mit Punkt "Sout" auf der Videointerface-Platine des Elekterminals zu verbinden, während Punkt 3 auf der Seite des Computers an "Sin" auf der Seite des Terminals liegen muß.
Wir hatten auch schon die Speisespannungsleitungen zwischen Computer und Terminal erwähnt. Diese sind natürlich nur notwendig, wenn das Terminal ohne eigene Stromversorgung arbeitet. Soll das Terminal universell verwendbar sein, dann ist ein eigenes Netzteil dringend zu empfehlen. Dient es jedoch ausschließlich der Kommunikation mit dem Junior-Computer, so kann man es aus dessen (erweitertem) Netzteil speisen und die beiden Speisespannungsleitungen (+5 V und −12 V) über zwei freie Konnektorkontakte (9 . . . 14) führen. Auf der Interfacekarte werden die betreffenden Konnektoranschlüsse über Drähte mit den nächstliegenden Platinenpunkten verbunden, die die benötigten Spannungen führen. Eine zusätzliche Entkopplung der beiden Speisespannungen durch Anlöten von zwei Tantalelkos 1 μ/16 V direkt am D-Konnektor (auf richtige Polarität der Elkos achten!) ist empfehlenswert.
In der Baubeschreibung für die Interfacekarte (Kapitel 10) ist angegeben, daß die Kontakte 4, 5 und 8 sowie 6 und 20 des dort sitzenden D-Konnektors über Drahtbrücken zu verbinden sind. Das ist beim Anschluß des Elekterminals an diesen Konnektor nicht notwendig, aber auch nicht schädlich. Diese Brücken sind nur erforderlich, wenn an den Junior-Computer ein peripheres Gerät (zum Beispiel ein großer Drucker mit integrierter Tastatur) angeschlossen wird, das eine vielseitigere Kommunikation mit dem Computer aufnehmen kann als das Elekterminal. Gemeint ist das sogenannte **Handshaking,** bei dem ein komplizierterer Austausch von Meldungen und Rückmeldungen zwischen Computer und Peripherie stattfindet. Da aber weder die Hardware noch die Software des Juniors für dieses "Händeschütteln" eingerichtet ist, müssen beim Anschluß eines derartigen Peripheriegeräts bestimmte Kommunikationskanäle kurzgeschlossen sein. Das periphere Gerät erhält dann jedesmal eine "künstliche" Rückmeldung, so daß die Aktivitäten ungehindert ablaufen können. In der Computersprache ausgedrückt: Wir müssen für ein "auto-loop back" sorgen, was durch Verbinden der Kontakte 4, 5 und 8 sowie 6 und 20 des RS-232-Konnektors auf der Interfacekarte geschah.

Nachdem das Elekterminal aufgebaut und angeschlossen wurde und die Art der Zusammenarbeit zwischen Terminal und Junior-Computer durchsichtig ist, wollen wir uns nun der wichtigsten Frage zuwenden: Was

können wir mit dem entstandenen System anfangen? Die Antwort auf diese Frage werden wir im folgenden Teil dieses Kapitels erfahren.

PRINTER MONITOR

Ein TV-(System-)Programm

Der Name "Printer Monitor" trifft eigentlich nur die Hälfte dessen, warum es hier geht. "Printer" bedeutet wörtlich "Drucker", so daß man unwillkürlich an das Drucken auf Papier, an lange, vom Computer ausgedruckte Listen mit auf den ersten Blick unverständlichem Inhalt denkt. Obwohl der Junior-Computer mit Hilfe des Systemprogramms PM derartiges auch zu Papier bringen kann, wird man ihn meistens auf andere Weise schreiben lassen: auf einen Bildschirm, der zum Beispiel Bestandteil des Elekterminals ist. Auch bei den bekannten, industriell gefertigten "Personal Computern" (PET, CBM, Apple usw.) wird diese Methode der visuellen Kommunikation des Computers mit dem Menschen bevorzugt. Eine treffendere Bezeichnung für PM wäre vielleicht "Video-Monitor" gewesen, doch hier entstehen leicht Gedankenverbindungen zu den TV-Überwachungsanlagen in Warenhäusern und Banken.
Das Systemprogramm PM kann als alternative, erweiterte Version des Standard-Monitors betrachtet werden. Mit PM bleibt ein Teil der vorangegangenen Aktionen und der darauf folgenden Reaktionen des Computers auf dem Bildschirm sichtbar. Ferner enthält PM eine Reihe von Tastenfunktionen, von denen zwei mit der Datenspeicherung auf Tonbandkassetten (Systemprogramm TM) in Zusammenhang stehen. An anderer Stelle in diesem Kapitel wurde bereits erwähnt, daß das Arbeiten in einer höheren Programmiersprache, zum Beispiel BASIC, auf der Basis von PM und der zugehörigen Hardware (Junior-Computer und Peripherie) möglich ist. Natürlich setzt das voraus, daß die Systemsoftware des Juniors durch ein Übersetzprogramm (Interpreter) erweitert wird.
Das Systemprogramm PM ist in einem EPROM vom Typ 2716 dauerhaft (resident) gespeichert; es braucht deshalb nicht nach jedem Einschalten des Junior-Computers erneut eingelesen zu werden. Wie aus Anhang 5 am Schluß dieses Buches hervorgeht, belegt PM den Adressenbereich 1000 . . . 14F3. Dieser Bereich umfaßt mehr als 1 KByte und weniger als 2 KByte, so daß im EPROM noch etwas Platz für zusätzliche, nicht flüchtige Software übrig bleibt. Sie läßt sich später durch Neuprogrammierung in das EPROM "einbrennen". Das PM-EPROM muß in die für IC5 vorgesehene Fassung auf der Interfaceplatine gesteckt werden; die Fassung für IC4 ist nicht für PM, sondern für TM bestimmt! Das Anpassen der Startadresse reicht für das Vertauschen von IC4 und IC5 nicht aus, es müßten zusätzlich alle im Programm vorkommenden absoluten Adressen modifiziert werden.
Ebenso wie TM kann auch PM vom Elektor-Programmier-Service in ein EPROM des Typs 2716 gesetzt werden. Wie man den Programmierservice in Anspruch nimmt, ist der neuesten Ausgabe der Zeitschrift "Elektor" zu entnehmen. Natürlich kann man das EPROM an Hand des Hexdump von PM (siehe Anhang 5) auch selbst programmieren, sofern man über ein

Programmiergerät verfügt. Eine Kontrolle des EPROM-Inhalts ist mit Hilfe des Standard-Monitors möglich: Startadresse von PM eingeben und anschließend durch Drücken der Taste + ein Byte nach dem anderen sichtbar machen.
Der Standard-Monitor enthält verschiedene Subroutinen, die auch in Anwenderprogramme eingebaut werden können. Dazu zählen Routinen für die Anzeige von Daten auf den Displays, das Abfragen der Tastatur sowie sogenannte Warteschleifen. Solche, für eigene Progamme nützliche Routinen, sind auch in PM enthalten: unter anderem eine Subroutine für das Schreiben von ASCII-Zeichen auf den Bildschirm oder auf Papier, für das Einlesen eines von der Tastatur gesendeten Zeichens (kontinuierliche Tastenabfrage) sowie mehrere grafische Subroutinen. Dazu gehört beispielsweise auch das Einschieben eines Zwischenraums und der Beginn einer neuen Zeile. Die Anwendungsmöglichkeiten dieser Subroutinen werden in Buch 4 behandelt, das sich unter anderem auch ausführlich mit der PM-Software beschäftigt.
Ein Anwenderprogramm kann unter der Regie von PM sowohl im "Computertempo" als auch Schritt für Schritt durchlaufen werden. Das war zwar schon mit dem Standard-Monitor möglich, doch nun läßt sich mit einem einzigen Tastendruck alles sichtbar machen, was über den Ablauf der einzelnen Programmschritte Aufschluß gibt.
Wir wollen nun die Einzelheiten der Anwendung von PM betrachten. Dazu besprechen wir zuerst an Hand von Bild 12 die verschiedenen Tastenfunktionen, die für den Umgang mit PM eine Rolle spielen.

① Taste RES = RUB OUT

Die Reservetaste RES der ASCII-Tastatur erhielt durch Ergänzung der Tastatur-Hardware die Funktion RUB OUT. Beim Drücken dieser Taste sendet die Tastatur über den UART das ASCII-Zeichen 7F (sieben mal logisch 1) zum Junior-Computer. Das löst den **zweiten** Start von PM aus. Um PM von Grund auf zu starten, wird der Standard-Monitor zu Hilfe genommen:
(Einschalten)
RST 1 Ø Ø Ø GO
Damit ist sozusagen der "PM-Motor" gestartet. Der Junior-Computer arbeitet einen Teil des Start-Vorprogramms von PM ab und wartet dann auf die Abfahrt des "PM-Zuges". Er setzt sich beim zweiten Startzeichen in Bewegung:
RUB OUT (= RES)
Das von der Tastatur gesendete **Startzeichen 7F** veranlaßt den Computer, den Rest des PM-Vorprogramms zu erledigen. Er stellt fest, mit welcher Bitgeschwindigkeit (Baudrate) das Zeichen 7F bei ihm ankommt. Alle später vom Computer rückwärts gesendeten Zeichen werden ebenso schnell gesendet wie empfangen. Nach Durchführung der dafür notwendigen Zeitmessungen meldet sich PM zurück mit dem Wort
JUNIOR
(Leerzeile)

Eine solche Rückmeldung heißt in der Computerfachsprache "Prompt". Auf den Text "JUNIOR" folgt zweimal automatisch das Kommando

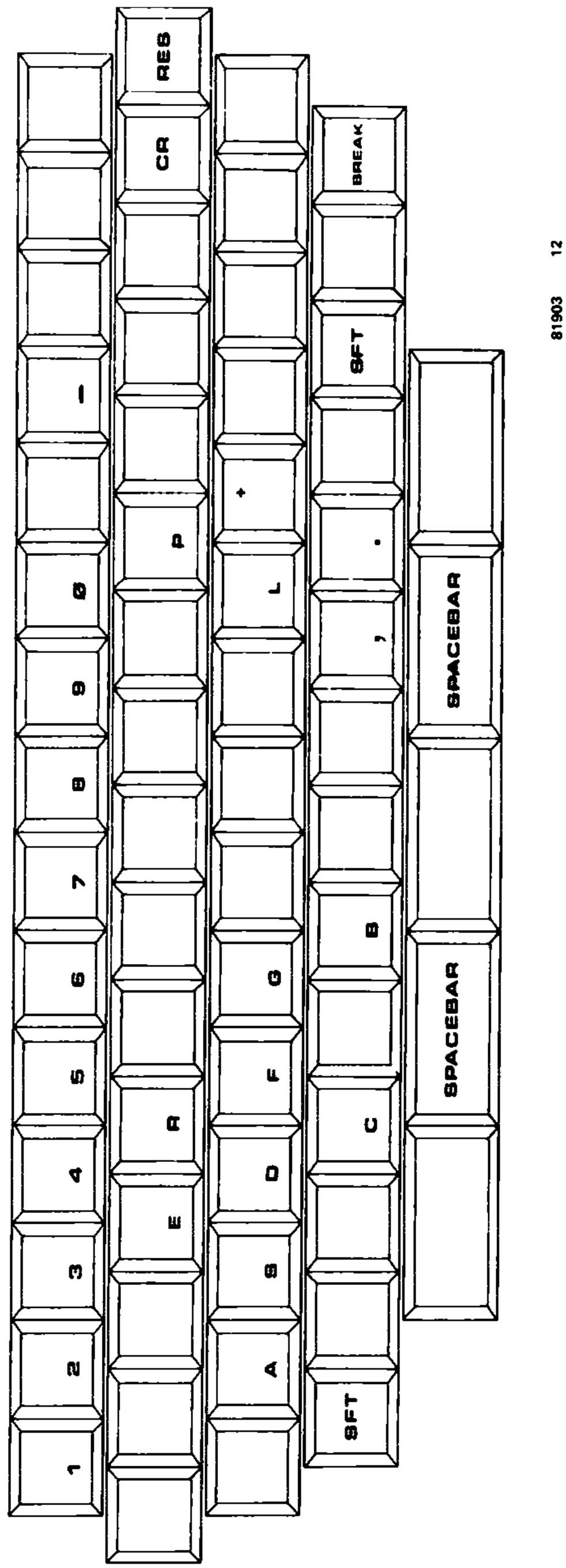


Bild 12. Die Tasten der ASCII-Tastatur, die für das Systemprogramm PM eine Bedeutung haben. Auch die Tastatur ist verkleinert wiedergegeben.

"neue Zeile" (CR = Carriage Return und LF = Line Feed), so daß zwischen dem Wort "JUNIOR" und dem später nachfolgenden Text eine Leerzeile entsteht.
Die Rückmeldung "JUNIOR" signalisiert die Betriebsbereitschaft von PM. Der Junior-Computer wartet auf einen Befehl in Form einer oder mehrerer Tastenbetätigungen.
Übrigens: Arbeitet man nicht mit dem Elekterminal, sondern mit einem anderen Datensichtgerät, so kann man das Startzeichen 7F in den meisten Fällen durch Drücken der Tasten "Control" und anschließend "DEL" (Delete) erzeugen.

② Tasten Ø . . . 9 und A . . . F
③ Taste SP (Zwischenraum)
④ Taste "." (Punkt)

Diese Tasten haben elementare Funktionen. Die numerischen Tasten Ø . . . 9 und A . . . F erfüllen die gleiche Aufgabe wie beim Arbeiten mit dem Standard-Monitor. Mit ihnen lassen sich Ziffern eingeben, die Bestandteile von Adressen, Speicherinhalten, Programmnummern usw. sein können. Für die Eingabe von Adressen und Daten sind spezielle Eingangspuffer vorhanden: die Speicherplätze INL und INH (siehe Tabelle 1). Das Eingeben einer **Arbeitsadresse** und das Ändern der dort stehenden Daten läuft mit PM anders ab als mit dem Standard-Monitor. Während dort die Tasten AD, DA, + und die numerischen Tasten gedrückt wurden, sind nun folgende Schritte notwendig:

- Zuerst wird die Ziffernfolge eingetippt; erst danach teilt man dem Computer mit, ob es sich um eine Arbeitsadresse oder um Daten handelt, die unter einer bestimmten Adresse in den Speicher zu setzen sind.
- Soll der Computer die eingegebene Ziffernfolge als Arbeitsadresse interpretieren, dann wird anschließend die Zwischenraumtaste SP gedrückt.
- Handelt es sich bei der Ziffernfolge um Daten, so drückt man Taste ".". In diesem Fall muß die Arbeitsadresse vorher eingegeben worden sein.
- Eine oder mehrere Vornullen von Adressen und Daten können bei der Eingabe wegfallen: Um Adresse Ø2ØØ anzusteuern, genügt die Eingabe von 2 Ø Ø. Drückt man ohne anderweitige Tastenbetätigung die Zwischenraumtaste, so ist ØØØØ die Arbeitsadresse. Ebenso wird die Arbeitsadresse mit ØØ geladen, wenn vor Drücken der Taste "." keine anderen Daten eingegeben wurden. Nach der Verarbeitung der Ziffernfolge werden die Eingangspuffer INL und INH wieder auf ØØ gesetzt, so daß die Vornullen auch beim nächsten und allen folgenden Eingabeschritten entfallen können.
- Nach der Eingabe von Daten (Taste ".") wird die Arbeitsadresse automatisch um eins erhöht und mit den zugehörigen Daten in der nächsten Zeile ausgedruckt. Es ist nicht erforderlich, wie beim Standard-Monitor eine Plustaste zu drücken. Dadurch werden bei der Eingabe von Programmen zahlreiche Tastenbetätigungen überflüssig, was die "Tipparbeit" nicht unwesentlich beschleunigt.

Wie die Eingabe eines Programms vor sich geht, wollen wir nun am Beispiel des Additionsprogramms aus Buch 1, Seite 63, erläutern. Zuerst spezifizieren wir für den BRK-Befehl den IRQ-Sprungvektor (er ist auf den SAVE-Eingang von PM gerichtet; darüber später mehr), danach geben wir

Tabelle 1. Liste der RAM-Speicherplätze, die in Zusammenhang mit dem Systemprogramm PM stehen.

```
TEMPORARY DATA BUFFERS

TEMP     *  $ØØFC
TEMPX    *  $ØØFD
STPBIT   *  $1A59   NUMBER OF STOP BITS
CNTL     *  $1A5A   BIT TIME BUFFER
CNTH     *  $1A5B
CNTHL    *  $1A5C   HALF BIT TIME BUFFER
CNTH     *  $1A5D
TIML     *  $1A5E   COUNT DOWN BUFFER
TIMH     *  $1A5F
TEMPA    *  $1AØØ   TEMPS
TEMPB    *  $1A61
CHA      *  $1A62   CHARACTER BUFFER
PARAL    *  $1A63   PARAMETER BUFFERS
PARAH    *  $1A64
PARBL    *  $1A65
PARBH    *  $1A66
PRTEMP   *  $1A67   TTY BUFFER
BRKT     *  $1A7C   BREAK TEST VECTOR

* * * BUFFERS & EXTERNAL ADDRESSES * * *

DISCNT   *  $1A68   DISPLAY COUNTER
COLDST   *  $1CB5   EDITOR COLD START
WARMST   *  $1CCA   EDITOR WARM START
BEGIN    *  $1ED3   EDITOR SUBROUTINE
RESET    *  $1C1D   RESET OF VERSION D
GETKEY   *  $1DF9   COMPUTE THE KEY VALUE
LDAINH   *  $1DA7   PART OF SCAND/SCANDS
BEGADL   *  $ØØE2   BEGIN ADDRESS POINTER
BEGADH   *  $ØØE3
ENDADL   *  $ØØE4   END ADDRESS POINTER
ENDADH   *  $ØØE5
CENDL    *  $ØØE8   CURRENT END ADDRESS POINTER
CENDH    *  $ØØE9
```

das Programm selbst ein. Nachdem PM gestartet ist und die Rückmeldung "JUNIOR" auf dem Bildschirm steht, drücken wir 1 A 7 E SP:

(1) 1A7E (SP)
(2) **1A7E XX**

Auf das Drücken der numerischen Tasten 1, A, 7 und E sowie der Zwischenraumtaste SP reagiert der Junior-Computer mit dem Ausdrucken der Arbeitsadresse auf einer neuen Zeile, einem Zwischenraum, der zur Arbeitsadresse gehörenden Daten und einem weiteren Zwischenraum. Die Tastenbetätigungen sind vor- und nachstehend in normalen Schrifttypen gesetzt, während die Reaktionen des Junior-Computers fett gedruckt wurden. Die am Anfang der Zeilen in Klammern stehenden Zahlen sind die Zeilen-

nummern; sie dienen nur der Übersichtlichkeit innerhalb dieses Buchtextes und erscheinen nicht auf dem Bildschirm.
Als nächster Schritt folgt die Eingabe der Daten CF unter der gewählten Arbeitsadresse:
(2) **1A7E XX** CF.
(3) **1A7F XX**
Nun ist die folgende Arbeitsadresse im Bild erschienen. Hier müssen die Daten 14 eingegeben werden:
(4) **1A7F XX** 14.
(5) **1A8Ø XX**
Die Adresse 1A8Ø ist für das Additionsprogramm bedeutungslos; die Eingabe des Programms muß mit Startadresse **ØIØØ** beginnen. Wir müssen deshalb zu dieser anderen Arbeitsadresse wechseln:
(5) **1A8Ø XX** IØØ (SP)
(6) **ØIØØ XX**
Danach wird das Additionsprogramm selbst eingegeben:
(6) **ØIØØ XX** 18.
(7) **ØIØI XX** A9.
(8) **ØIØ2 XX** 13.
(9) **ØIØ3 XX** 69.
(10) **ØIØ4 XX** Ø8.
(11) **ØIØ5 XX**
(12) **ØIØ6 XX**
In Speicherplatz ØIØ5 wurde buchstäblich ein Schlußpunkt gesetzt: Die letzte Instruktion des Additionsprogramms ist der BRK-Befehl mit dem Opcode ØØ.

⑤ Taste "+"
⑥ Taste "–"

Taste "+" erhöht die Arbeitsadresse um eins, während Taste "–" die Arbeitsadresse um eins herabsetzt. In beiden Fällen wird nach der Tastenbetätigung die neue Arbeitsadresse mit den zugehörigen Daten ausgedruckt. Mit Hilfe dieser beiden Tasten lassen sich die Inhalte von kleineren oder auch größeren Speicherabschnitten überprüfen, wie zum Beispiel der Adressenbereich, in dem das Additionsprogramm steht:
(12) **ØIØ6 XX** +
(13) **ØIØ7 XX**
Pardon, wir mußten natürlich die Taste "–" und nicht "+" drücken:
(13) **ØIØ7 XX** –
(14) **ØIØ6 XX** –
(15) **ØIØ5 ØØ** –
(16) **ØIØ4 Ø8** –
(17) **ØIØ3 69** –
(18) **ØIØ2 13** –
(19) **ØIØI A9** –
(20) **ØIØØ 18**
Damit sind wir wieder bei der Startadresse angelangt. Es wurde alles richtig eingegeben, so daß das Programm gestartet werden kann.
Noch eine wichtige Anmerkung: Das Auslösen der Tastenfunktion "+" macht auf dem Elekterminal das gleichzeitige Drücken der Umschalttaste SFT (Shift) erforderlich.

⑦ Taste R

Das R steht für RUN, was hier soviel wie "Programmstart" bedeutet. Diese Taste entspricht in ihrer Funktion der GO-Taste auf der Basisplatine: Wird sie gedrückt, dann arbeitet der Computer das Programm ab, das bei der gewählten Arbeitsadresse beginnt. Ob dies in langsamen Einzelschritten oder mit Computergeschwindigkeit geschieht, hängt von der Stellung des STEP-Schalters ab.
Und so starten wir das Additionsprogramm:

(21) **0100 18** R
(22) **0107 XX**

Wie kommt Zeile 22 zustande? Der IRQ-Sprungvektor ist auf Adresse 14CF gerichtet; diese Adresse hat für PM die gleiche Bedeutung wie Adresse 1C00 für den Standard-Monitor. Im Fall eines NMI oder IRQ durchläuft der Prozessor ein zu PM gehörendes Interrupt-Vorprogramm, das verschiedene Programmdaten rettet. Gleichzeitig setzt das Vorprogramm die Adresse mit den zugehörigen Daten auf den Bildschirm, die ohne Interrupt als nächste an der Reihe gewesen wäre (Stand von Programmzähler PC). Bei BRK tritt der bekannte "Bremsweg" auf: eine Adresse wird überschlagen, so daß in Zeile 22 Adresse 0107 an Stelle von 0106 erscheint.

⑧ Taste P

Die Funktion dieser Taste entspricht der Funktion von Taste PC des Standard-Monitors. Beim schrittweisen Durchlaufen eines Programms (Step-Mode) sorgt P dafür, daß nach Ausstieg aus dem Programm (um die einzelnen Ablaufphasen untersuchen zu können) der für die Fortsetzung notwendige "rote Faden" nicht verloren geht. Die Rückkehradresse wird nach Drücken von P auf einer neuen Zeile ausgedruckt; sie ist die neue Arbeitsadresse. Der Prozessor führt die nächste Instruktion aus, sobald anschließend Taste R gedrückt wird.

⑨ Taste L

L ist die Abkürzung des englischen Worts "List". Gemeint ist damit eine Liste der Inhalte aller internen 6502-Register. Diese Registerinhalte werden während des Interrupt-Vorprogramms von PM in RAM-Speicherplätze gesetzt. Wie diese Liste aussieht, läßt sich am besten an einem Beispiel zeigen. Wir setzen unsere Betrachtung bei Zeile 22 fort und drücken L:

(22) **0107 XX** L
(23) **ACC: 1B**
(24) **Y: XX**
(25) **X: XX**
(26) **PC: 0107**
(27) **SP: 01FF**
(28) **PR: 00 1 1 0 1 0 0**
(29) **NV B D I Z C**

Auf das C in Zeile 29 folgt ein Zwischenraum.
Die Registerinhalte des Prozessors lassen sich, wie das Beispiel zeigt, mit PM wesentlich schneller und übersichtlicher darstellen als mit dem Standard-Monitor. Auf einen Blick erfährt man alles Wissenswerte und braucht nicht nacheinander die Speicherplätze 00EF . . . 00F5 abzufragen.

Der Inhalt des P-Registers erscheint bitweise auf dem Schirm; darunter sind sogar die Abkürzungen der einzelnen Flags ausgedruckt. In diesem Fall steht im Akku das Ergebnis der Addition (1B). Das X- und das Y-Register spielen beim Additionsprogramm keine Rolle. Der PC hat infolge des BRK den Inhalt Ø1Ø7. In Zeile 27 ist der Inhalt von SP (SPUSER) ausgedruckt. Dieser ist ebenso wie beim Start von PM Ø1FF, da wir den Stack nicht benutzt haben und auch nicht das Programm innerhalb einer Subroutine unterbrachen. Die B-Flag ist 1, weil der Prozessor auf ein BRK stieß, und die Flags N, Z und C sind sämtlich Ø, weil das Additionsergebnis positiv und kleiner als FF ist.

⑩ Taste M
⑪ Taste ","
⑫ Taste CR

Das M bedeutet so viel wie "Matrix"; darunter versteht man in diesem Zusammenhang eine Tabelle, die Daten enthält und aus mehreren Reihen und mehreren Spalten besteht. Auch ein Hexdump ist eine Datenmatrix, wobei die Anzahl der Dateneinheiten pro Zeile auf maximal 16 begrenzt ist; dies ist gleichzeitig die Anzahl der Spalten. Die Zahl der Reihen hängt von der Datenmenge ab. Teilt man die Anzahl der Dateneinheiten durch 16 und zählt, sofern sich ein Rest ergibt, eins hinzu, so erhält man die Zeilenzahl. Die letzte Zeile kann unvollständig sein, so wie zum Beispiel diese Buchtextzeile.

Zusammen mit den Tastenfunktionen S und G gehört M zur Gruppe der **parametrisierten Tastenfunktionen:** Das Drücken der Tasten M, S oder G reicht allein nicht aus, sondern der Computer benötigt zum Ausführen des Tastenbefehls weitere Informationen. Diese Informationen heißen **Parameter;** sie müssen nach Drücken von M, S oder G zusätzlich eingegeben werden. Besteht diese Information aus mehr als einer Dateneinheit, dann werden die Dateneinheiten durch ein Komma voneinander getrennt. (IRQ – ein sprachliches Problem: Wie lautet die Einzahl von Daten? Vielleicht Datum? Oder, wenn "Data" die Einzahl ist, müßte die Mehrzahl dann nicht "Datas" heißen? Oder vielleicht "Datassen"?! – RTI.) Nach dem ersten Parameter folgt also ein ",", daran schließt sich der zweite Parameter an usw. Der Computer führt den so eingegebenen Tastenbefehl aus, wenn Taste "CR" (Carriage Return = Zurück zum Zeilenanfang) gedrückt wird.

Welche Parameter spielen beim Ausdrucken eines Hexdump eine Rolle? Es sind genau zwei: die erste und die letzte Adresse des Datenblocks, der als Hexdump sichtbar gemacht werden soll. Auch hier nehmen wir zur Verdeutlichung das Programmbeispiel zu Hilfe. Wir waren auf Zeile 29 stehengeblieben:

(29) **NV BDIZC** M
(30) **HEXDUMP:**

Wir geben zuerst die Anfangsadresse des Datenblocks ein, drücken ",", lassen die letzte Datenblockadresse folgen und betätigen dann Taste CR. Die links stehenden Nullen (Vornullen) können auch hier wegfallen; das Komma signalisiert dem Computer, wann die erste Adresse vollständig ist. Die Anfangsadresse des auszudruckenden Datenblocks muß selbstverständlich niedriger liegen als die Endadresse. Anderenfalls reagiert der Computer mit einer Fehlermeldung, über die später noch einiges anzu-

merken ist. Wir lassen jetzt das Additionsprogramm ausdrucken:
(30) **HEXDUMP**: 1ØØ, 1Ø5 CR
(31) **Ø 1 2 3 4 5 6 7 8 9 A B C D E F**
(32) **Ø1ØØ: 18 A9 13 69 Ø8 ØØ**
(33) **(Leerzeile)**
(34) **JUNIOR**
(35) **(Leerzeile)**
(36)
Wie wir sehen, wird als erstes automatisch eine Zeile mit den Ziffern der Spalten (Ø . . . F) ausgedruckt. Die folgende Zeile beginnt mit der Adresse, zu der die erste Dateneinheit in dieser Zeile gehört. Das Hexdump des Additionsprogramms füllt wegen seiner Kürze noch nicht einmal eine einzige Zeile aus. Wäre der auszudruckende Datenblock umfangreicher, dann würden zwischen den Zeilen 32 und 33 entsprechend viele Zeilen in Form von Zeile 32 stehen.
Die Anfangsadressen der Zeilen müssen nicht unbedingt eine Ø als letzte Ziffer haben, sondern das Hexdump kann mit jeder beliebigen Adresse beginnen. Trotzdem ist es zweckmäßig, als Anfangsadresse XXXØ (mit X = Ø . . . F) zu wählen, weil dann die Zugehörigkeit der Daten zu den jeweiligen Adressen auf den ersten Blick zu übersehen ist. Auch für das Ausdrucken des Additionsprogramms (Zeilen 31 und 32) wurde diese Darstellungsart gewählt.

⑬ Taste S

Diese Tastenfunktion ist mit der Tastenfunktion SAVE des Systemprogramms TAPE MANAGEMENT (siehe Kapitel 11) identisch. Hiermit können Datenblöcke auch während des Verbleibs in PM auf Tonbandkassetten gesetzt werden. Allerdings wird dazu die Subroutine DUMP von TM benutzt, so daß Tastenfunktion S nur bei Vorhandensein von TM zur Verfügung steht.
Auch in diesem Fall handelt es sich um eine parametrisierte Tastenfunktion: Nach Drücken von Taste S müssen nacheinander die Programmnummer ID, die Startadresse SA und die Endadresse EA (= LA + 1) eingegeben werden. Während bei TM die Tastenfunktion sofort beim Drücken von Taste GET ausgeführt wurde, geschieht dies hier erst, wenn nach Eingabe der Parameter Taste CR gedrückt wird. Die für die Datenkonservierung auf Band sonst noch notwendigen Vorbereitungen wurden bereits in Kapitel 11 beschrieben.
Die Programmnummern ØØ und FF sind für die Tastenfunktion S als Parameter nicht zugelassen. Gibt man sie trotzdem ein, dann folgt eine Fehlermeldung. Wir setzen unsere Betrachtung am Beispiel des Additionsprogramms fort:
(36) S11, 1ØØ, 1Ø6 CR
(37) **READY**
(38)
Zeile 37 erscheint auf dem Bildschirm, sobald der Datenblock vollständig auf dem Band steht; nachfolgende Aktivitäten werden auf Zeile 38 sichtbar gemacht. Das Additionsprogramm erhielt hier die Programmnummer 11. Auf das Band wurden sämtliche Daten von SA bis EA (einschließlich LA und ausschließlich EA) geschrieben.

⑭ Taste G

Wie Taste S ihr Äquivalent in Taste SAVE von TM hat, so entspricht Taste G der Taste GET. Taste G wird gedrückt, wenn ein Datenblock von einer Tonbandkassette gelesen werden soll. Auch für G ist das Vorhandensein von TM notwendig, da der Computer die Daten mit Hilfe der zu TM gehörenden Subroutine RDTAPE vom Band liest. Ein Parameter, der in jedem Fall eingegeben werden muß, ist die Programmnummer ID. Wählt man FF als Programmnummer, dann ist außerdem die Startadresse SA einzutippen. Bei ID = FF wird nämlich der auf dem Band nächstfolgende Datenblock ohne Rücksicht auf die dort stehende Programmnummer in das RAM gelesen. Der RAM-Bereich beginnt dabei mit Startadresse SA (vergleiche Kapitel 11).
Auch hier wieder das Beispiel aus der Praxis. Das Additionsprogramm erschien uns so wertvoll, daß wir beschlossen hatten, es auf Band zu setzen (Zeilen 36 und 37). Wir brauchen uns nun nicht mehr die Mühe zu machen, die sechs Byte nach Abschalten und Wiedereinschalten des Junior-Computers erneut einzutippen. Das Programm wird jetzt vom Band gelesen:
(38) G11CR
(39) **READY**
Die Rückmeldung READY erscheint auf dem Bildschirm, sobald der Datenblock mit der Programmnummer 11 gefunden und in den RAM-Bereich Ø1ØØ . . . Ø1Ø6 geschrieben ist. Nun kann der Fall eintreten, daß wir den RAM-Bereich ab Adresse Ø1ØØ für andere Zwecke brauchen, jedoch gleichzeitig das Additionsprogramm benutzen wollen. Letzteres muß deshalb in einen anderen Bereich, zum Beispiel auf Seite ØØ ab Adresse ØØ1Ø umgesetzt werden. Das Band wird bis zu einer Stelle kurz vor Beginn der Datensendung 11 zurückgespult, der Rekorder wird unmittelbar nach Zeile 41 gestartet:
(40) GFF
(41) **SA:** 1ØCR
(42) **READY**
Das Additionsprogramm wurde in den RAM-Bereich ØØ1Ø . . . ØØ16 kopiert. Kontrollieren läßt sich dies durch Ausdrucken der in diesem Speicherabschnitt stehenden Daten mit Hilfe von Tastenfunktion M.

⑮ Taste BRK

Nehmen wir an, daß wir dem Computer den Befehl erteilt haben, den Inhalt des Speicherbereichs ØØØØ . . . FFFF auszudrucken. Kurz darauf stellen wir fest, daß uns nur der Bereich ØØØØ . . . ØØØF interessiert. Das Ausdrucken des ersten Bereichs nimmt 4096 mal mehr Zeit in Anspruch als das Ausdrucken des zweiten Bereichs. Eine Möglichkeit der Unterbrechung, auf die ein Sprung zu einer definierten Adresse (z.B. Startadresse einer geeigneten Subroutine) folgt, ist deshalb äußerst nützlich. Ziehen wir kurzerhand den Netzstecker heraus? Das wäre eine schlechte Lösung. Taste RST drücken, also zurück zum Standard-Monitor? Diese Lösung ist nicht viel besser. Helfen würde uns eine ähnliche Tastenfunktion von PM, wie sie der Standard-Monitor mit ST besitzt. ST löst bekanntlich einen NMI aus, auf den ein indirekter Sprung (JMP-IND) zu

den Adressen 1A7A und 1A7B folgt. Dort steht der meistens auf den Standard-Monitor (1CØØ) gerichtete, aber auch beliebig anders modifizierbare NMI-Sprungvektor.
Die Taste BRK schafft innerhalb von PM ähnliche Möglichkeiten, die allerdings unabhängig vom NMI-Mechanismus ablaufen. Aus der Teilschaltung des Elekterminals (Bild 5) geht hervor, daß der Kontakt von Taste BRK auf der einen Seite an Masse liegt (lies: liegen muß) und auf der anderen Seite mit dem gepufferten s.o.-Ausgang der UART verbunden ist. Drückt man BRK, so wird die im Ruhezustand am s.o.-Ausgang und an Port PA7 des Junior-Computers liegende logische 1 auf logisch Ø herabgezogen. Die Spannung auf der zugehörigen RS-232-Leitung springt dann von −12 V auf +5 V.
TM enthält eine Subroutine für das Senden der ASCII-Zeichen vom Computer zur Peripherie: die Subroutine PRCHAR. Während der letzten Ausführungsphase dieser Subroutine wird geprüft, ob Portleitung PA7 logisch 1oder logisch Ø ist. Bei PA7 = log. Ø wartet der Computer auf den Wechsel nach logisch 1 (Loslassen der BRK-Taste) und springt dann indirekt zu der als **BRK-Sprungvektor** spezifizierten Adresse. Der BRK-Sprungvektor steht auf den Speicherplätzen
1A7C (niedriges Byte) und
1A7D (höheres Byte).
Unmittelbar nach dem Start von PM ist der BRK-Sprungvektor auf das Label LABJUN mit der Adresse 1Ø5F gerichtet. Das dort beginnende Programm sorgt für die Rückmeldung "JUNIOR" auf dem Bildschirm und das Setzen auf FF von Stackpointer SP und Speicherplatz SPUSER (Adresse ØØF2). Das Spezifizieren eines anderen BRK-Sprungvektors ist möglich; dies muß jedoch während des Verbleibs in PM geschehen. Der BRK-Sprungvektor des Systemprogramms PM darf übrigens nicht mit dem IRQ-Sprungvektor verwechselt werden, der bei einem BRK-**Befehl** maßgebend ist!
Noch eine Anmerkung: Aus dem Vorstehenden wird deutlich, daß die BRK-Taste nur wirksam ist, wenn der Junior-Computer ASCII-Zeichen sendet. Solange PM beispielsweise auf das Drücken einer Taste wartet, hat Taste BRK keinen Einfluß.

Damit wurden alle Tastenfunktionen des Systemprogramms PM besprochen. Drückt man auf der ASCII-Tastatur eine der übrigen Tasten, denen keine Funktion innerhalb von PM zugeordnet ist, dann erscheint auf dem Bildschirm die Frage "WHAT"? Oder PM antwortet mit "JUNIOR", was aber auch keine neuen Erkenntnisse liefert. Im folgenden Abschnitt dieses Kapitels wollen wir sehen, wozu die Tasten R, P und L außerdem noch verwendbar sind.

Schritt für Schritt mit PM

Ausdrucken von Momentaufnahmen

Auch mit PM kann man ein Programm in Einzelschritten durchlaufen. Dazu muß die Hardware der Basisplatine wie in Kapitel 10 beschrieben angepaßt worden sein. Das betrifft vor allem die Aufsteckplatine für IC10;

siehe hierzu die Bilder 8b, 9, 11 und 12 in Kapitel 10. Die beiden K-Leitungen der Aufsteckplatine wurden so mit den Anschlüssen von IC6 verbunden, daß Programme im Adressenbereich 1ØØØ . . . 17FF (K4 und K5; diese Anschlüsse sind wegen des PM-EPROMs miteinander verbunden) sowie im Adressenbereich 1CØØ . . . 1FFF (K7) **nicht** schrittweise durchlaufen werden können. Wegen einer im PIA-RAM auf Seite 1A stehenden Monitor-Ergänzung ist in einem bestimmten Fall zusätzlich eine NMI-Blockierung des durch K6 selektierten Adressenbereichs notwendig. Das betrifft sowohl den Standard-Monitor (K7 und K6; siehe Anhang 2) als auch PM (K4, K6 und K7), wenn der Computer dezimal rechnen soll (siehe nachfolgend in diesem Kapitel).
Und so lassen wir den Computer mit PM in Einzelschritten arbeiten:

- Der STEP-Schalter S24 wird auf ON gestellt, so daß die rote LED in der GO-Taste aufleuchtet.
- Nicht erforderlich ist das Spezifizieren des NMI-Sprungvektors (außer beim dezimalen Rechnen). Dieser wird nämlich beim Start von PM automatisch auf Adresse 14CF mit dem Label STEP gerichtet. Dort beginnt eine Art SAVE-Routine: Die Inhalte der Prozessor-Register werden in die bekannten Speicherplätze auf Seite ØØ gesetzt. Ferner wird die Adresse mit den zugehörigen Daten ausgedruckt, unter der der Opcode der nächsten auszuführenden Instruktion steht.

Ein praktisches Beispiel

In Kapitel 3 (Buch 1) wurde das schrittweise Durchlaufen von Anwenderprogrammen mit dem Standard-Monitor beschrieben. Mit PM läuft diese Prozedur ähnlich ab, so daß wir im Bedarfsfall auf die dort gegebenen grundsätzlichen Erläuterungen zurückgreifen können.
Hier nun das Programmbeispiel. Wir betrachten folgende Zahlenreihe:

0	1	3	6	10	15	.	.	.	(Zahlenreihe)
	1	2	3	4	5	.	.	.	(Differenzen)

Die Differenz zwischen der ersten und der zweiten Zahl ist 1, zwischen der zweiten und der dritten Zahl ist sie 2, zwischen der dritten und der vierten Zahl beträgt die Differenz 3, und so weiter. Wir schreiben ein Programm, das die ersten elf Zahlen dieser Reihe einschließlich der am Anfang stehenden Null berechnet. Die Glieder der Reihe (die berechneten Zahlen) sollen in der richtigen Reihenfolge mit der Zahl Null auf Speicherplatz Ø1FF beginnend in den Stack gesetzt werden. Der Nutzen eines solchen Programms soll außer Betracht bleiben; es dient ausschließlich zur Verdeutlichung der "Step-Praxis".
Das Programm, das wir NUMBERS nennen, ist in Bild 13 zu finden. Es ist wie folgt aufgebaut: Zuerst wird Y auf den Anfangswert Ø1 gesetzt und die erste Zahl (ØØ) auf den Stack geschoben. Danach durchläuft der Prozessor zehn mal hintereinander den beim Label ADD beginnenden Programmteil. Der Wert von Y wird in Speicherplatz TEMPY abgelegt, die zuvor berechnete Zahl wandert in den Akku (PLA und PHA), und zum Inhalt des Akkus wird der Wert von Y (Inhalt von TEMPY) addiert. Die Berechnung der nächsten Zahl ist damit ausgeführt; sie wird auf den Stack geschoben (PHA). Dann wird Y um eins erhöht (INY) und anschließend die nächste Zahl berechnet. Ein Vergleich (CPY #ØB und BNE ADD) entscheidet, ob

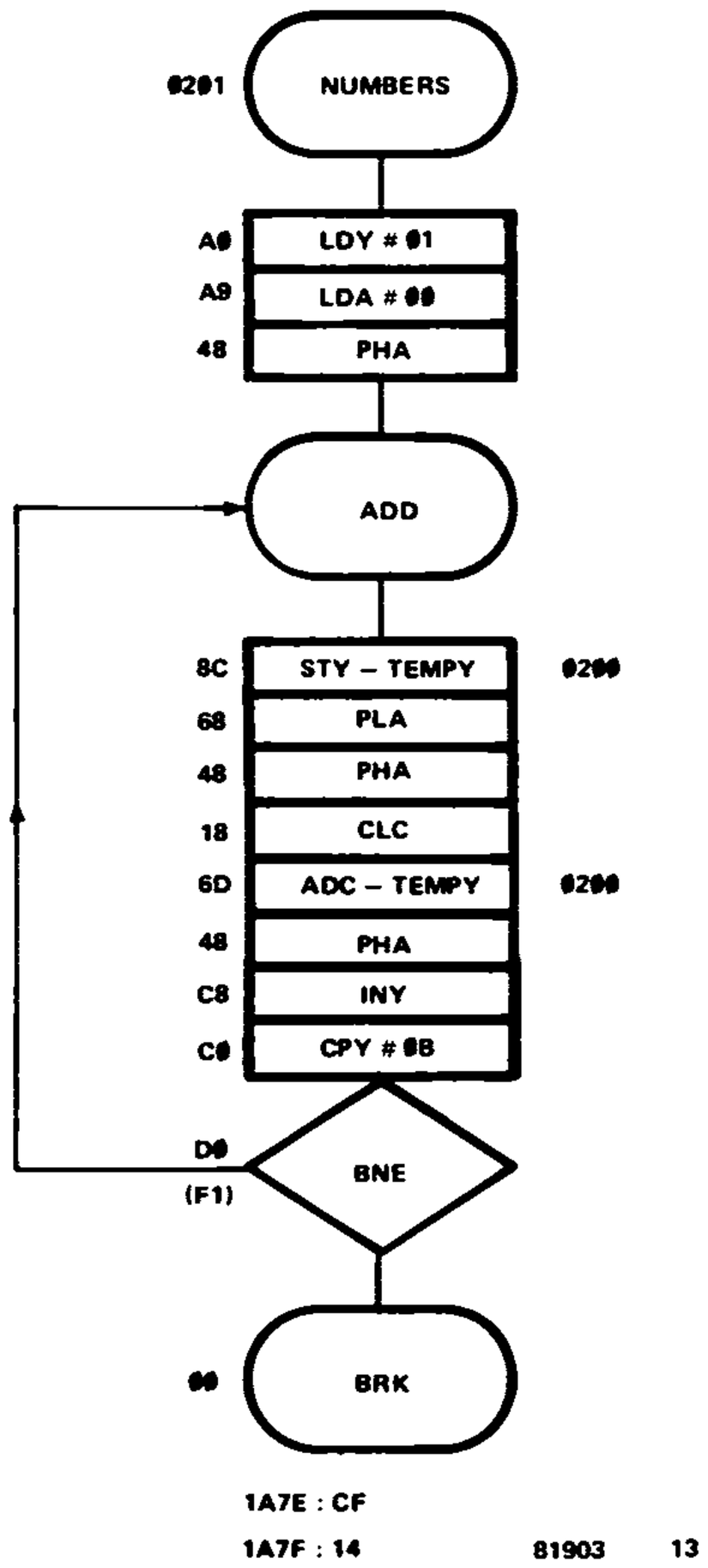


Bild 13. Mit dem Programm NUMBERS kann eine Zahlenfolge in de Stack gesetzt werden, deren Glieder nach einer bestimmten Gesetzmäßigkeit wachsen.

die nächste Zahl noch zu berechnen ist (Sprung nach ADD) oder nicht (über BRK zum STEP/SAVE-Eingang von PM).
In Bild 14 sind der Stack und der Stackpointer für verschiedene Werte von Y skizziert. Der Inhalt der Speicherplätze Ø1FF . . . Ø1F5 bezieht sich auf die binäre (hexadezimale) Rechenweise, mit der auch das Programm in Bild 13 arbeitet. Durch Rechnen im Dezimalsystem ergeben sich die Inhalte der Speicherplätze Ø1F6 . . . Ø1EA; hierzu später mehr.
Um das schrittweise Durchlaufen des Programms in Bild 13 weiter zu verdeutlichen, haben wir dieses Programm in Bild 15 auf etwas ungewöhnliche Art dargestellt. Die schwarzen Kugeln symbolisieren die Speicherplätze, in denen ein Opcode steht. Das sind die Zwischenstationen, bei denen der fahrende Zug (der Programmablauf) eine Pause einlegt. Während der Zug beim schrittweisen Durchlaufen des Programms als "Bummelzug" fährt, hält er im Normalbetrieb nur ein einziges Mal an: bei DECAR mit

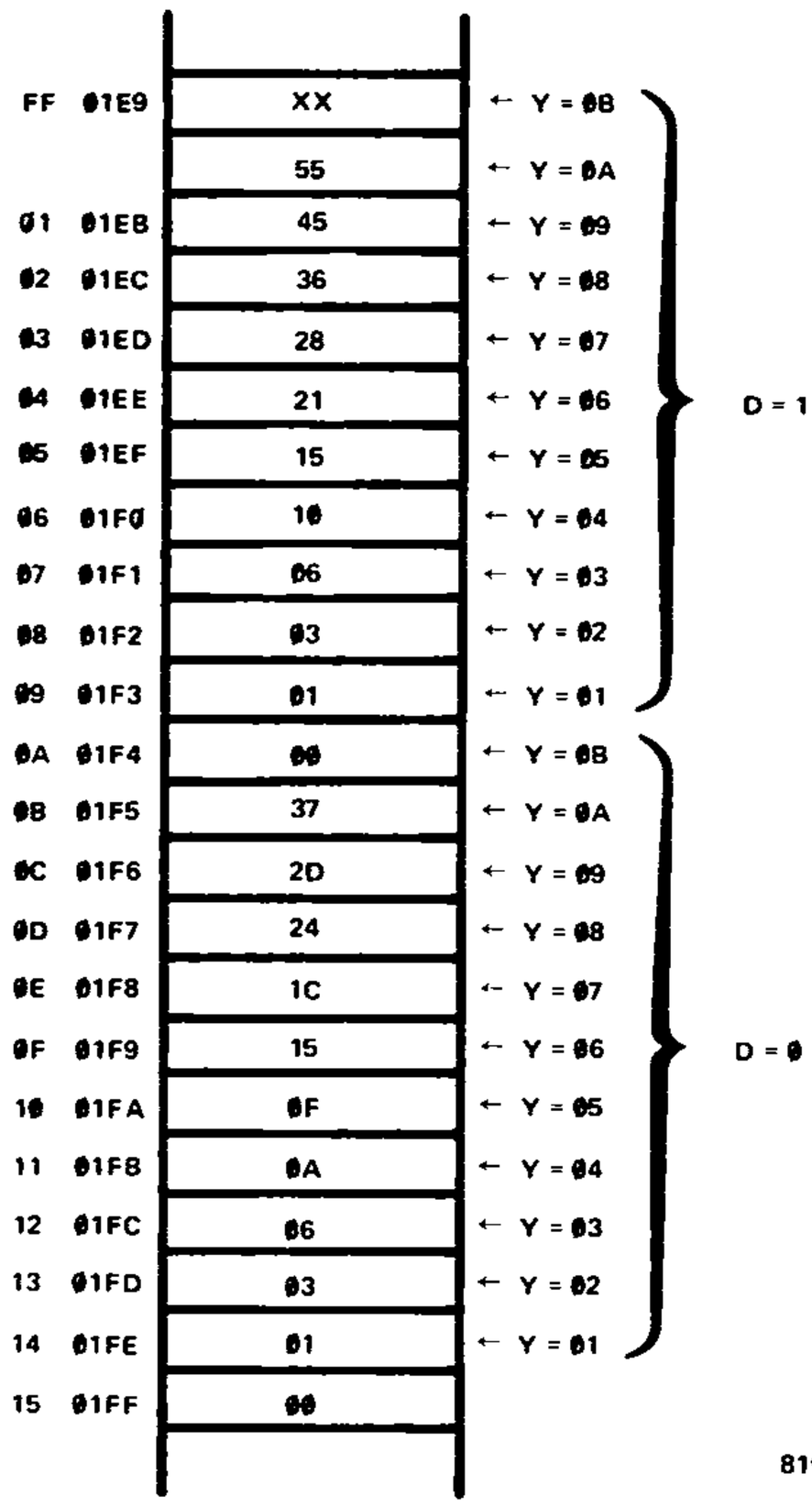


Bild 14. Der Inhalt des Stack nach Durchlaufen von NUMBERS im binären und dezimalen Rechenmodus. Die rechts angegebenen Y-Werte beziehen sich auf NUMBERS (Bild 13), die links stehenden auf PRNUMB (Bild 18).

der Adresse Ø217. Der letzte Abschnitt der Reise ist der BRK-Bremsweg, der vom BRK-Befehl in Adresse Ø215 ausgelöst wird.
Das Programm NUMBERS haben wir in Gestalt des detaillierten Flußdiagramms (Bild 13) sowie als "Zuglaufplan" kennengelernt. Eine dritte, vertrautere Form ist die Assembler-Schreibweise:

```
NUMBERS  0201  A0  01        LDY #01
         0203  A9  00        LDA #00
         0205  48            PHA
ADD      0206  8C  00  02    STY-TEMPY
         0209  68            PLA
         020A  48            PHA
         020B  18            CLC
         020C  6D  00  02    ADC-TEMPY
         020F  48            PHA
         0210  C8            INY
         0211  C0  0B        CPY #0B
```

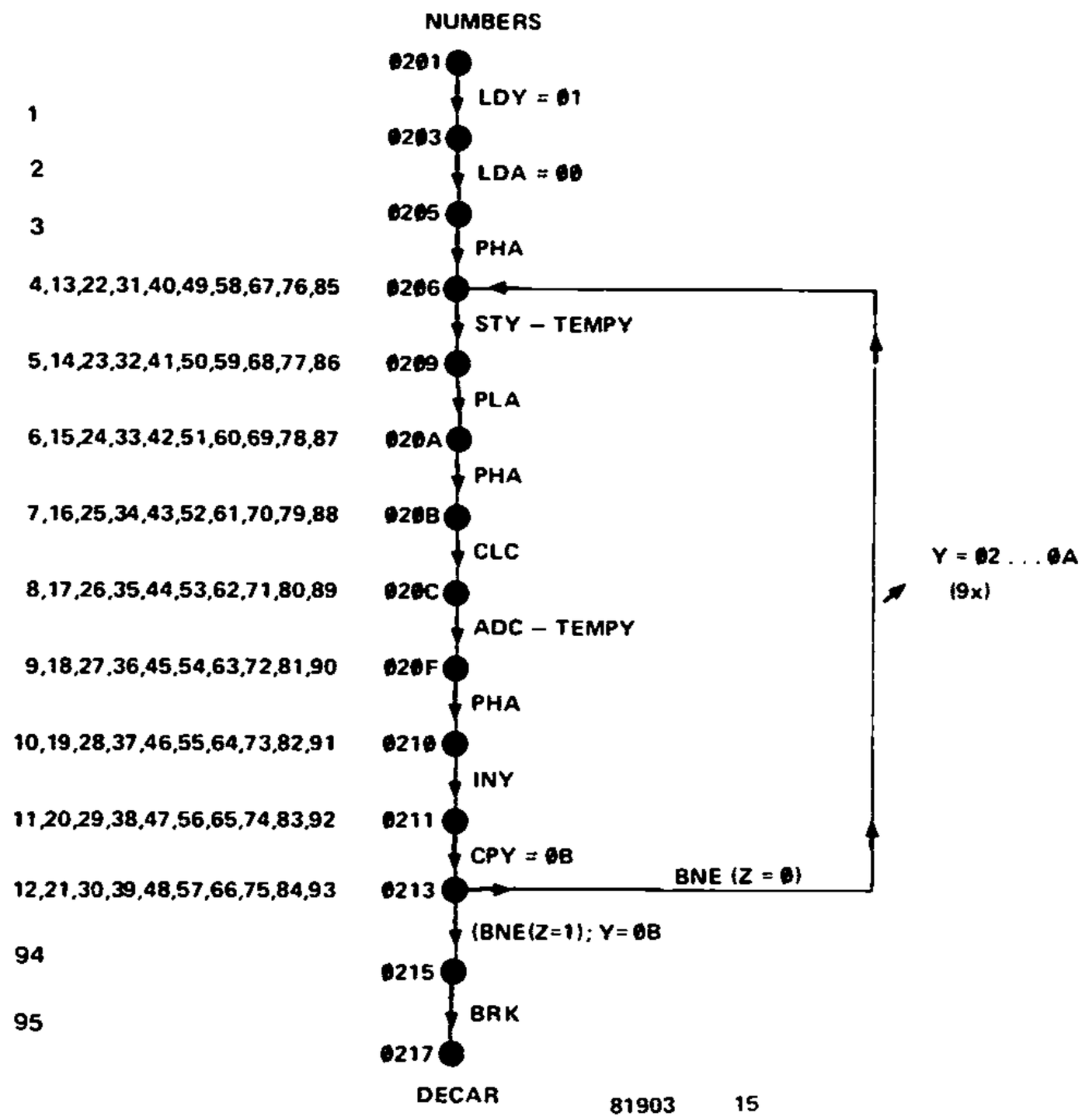


Bild 15. Das "Schienen-Modell" für das Programm NUMBERS aus Bild 13. Die Zahl der Stationen ist um eins größer als die Anzahl der Instruktionen von NUMBERS. Stopstellen sind jedoch insgesamt 94 vorhanden, weil die Stellung der Weiche bei Station 0213 von der Z-Flag abhängt.

```
        0213  D0  F1     BNE ADD
BRK     0215  00         BRK
        1A7E  CF
        1A7F  14
```

Die erste Spalte ist hier für die Label bestimmt, in der zweiten stehen die Adressen der Speicherplätze, die einen Opcode enthalten (mit Ausnahme des IRQ- Sprungvektors am Ende). Dann folgen die Bytes, jeweils eine Instruktion pro Zeile, und schließlich die Mnemonics. In die letzte Spalte können noch Kommentare und Erläuterungen gesetzt werden.

Für das schrittweise Durchlaufen des Programms NUMBERS zählen wir 93 Zwischenstationen ohne die Endstation bei DECAR. Einschließlich des Ausgangs- und des Zielpunktes sind somit 95 unterschiedliche Momentaufnahmen des Programmablaufs möglich. Diese Momentaufnahmen setzen sich aus dem Inhalt von Speicherplatz TEMPY (Adresse 0200) und der verschiedenen Prozessor-Register zusammen. An dieser Stelle sollen nicht sämtliche 95 Momentaufnahmen wiedergegeben werden, denn das würde zu viele Buchseiten (bzw. TV-Schirm-Seiten) füllen. Wir beschränken uns deshalb auf die Wiedergabe einiger weniger Momentaufnahmen.

Zuvor muß jedoch erst einmal PM gestartet und das Programm NUMBERS eingetippt werden:
RST 1 Ø Ø Ø GO RUB OUT
JUNIOR
(Leerzeile)

Was unser Drucker danach an "Momentaufnahmen" produziert hat, das ist original in Tabelle 2 wiedergegeben. Zuerst tippten wir die verschiedenen Daten ein, dann wurde wieder die Startadresse aufgesucht; der STEP-Schalter steht dabei auf ON. Die Stationsnummern 1 bis 95, die in der "Eisenbahnversion" des Programms NUMBERS (Bild 15) vorkommen, finden sich im "Listing" von Tabelle 2 wieder. Diese Zahlen wurden nicht von PM ausgedruckt, sondern später aus Gründen der Übersichtlichkeit hinzugefügt.
Festgehalten wurden die Inhalte der Prozessor-Register (durch Drücken von L) und des Speicherplatzes TEMPY (durch Drücken von 2 Ø Ø SP) an den Punkten 1 (unmittelbar vor dem Start von NUMBERS) und 95 (nach dem abschließenden BRK; unabsichtlich wurden einige willkürliche Instruktionen ausgeführt) sowie an den Punkten 12, 23, 30, 39, 48, 57, 66, 75, 84 und 93: den Zwischenstationen mit "BNE-Weichen" (ausgenommen Nummer 23, wir haben Taste R einmal zu viel gedrückt). Zum Schluß erscheinen die Speicherplätze mit ihrem Inhalt, in die die vom Programm NUMBERS berechneten Zahlen gesetzt wurden. Übrigens: Um Platz zu sparen, wurde Tabelle 2 hier in Form von drei Spalten pro Buchseite abgedruckt.
Die softwaremäßigen Zusammenhänge zwischen den einzelnen Register- und Speicherinhalten während der verschiedenen Programmphasen sollen an dieser Stelle nicht untersucht werden; nichts hindert jedoch daran, dies selbst zu tun. Wir wollen lediglich zeigen, daß das Systemprogramm PM des Junior-Computers mit den Tastenfunktionen L und P über äußerst nützliche Werkzeuge verfügt, die den Programmablauf schnell und vollständig überschaubar machen. Die anschließende Analyse des Geschehens wird meistens wesentlich mehr Zeit in Anspruch nehmen, vor allem, wenn die Entwicklung des Programms noch wenig fortgeschritten ist.

Tabelle 2. Die folgenden Seiten geben ein PM-Listing (Hard Copy) wieder, das beim zweimaligen, schrittweisen durchlaufen des Programms NUMBERS aus Bild 13 entstand. Auch die Dateneingabe ist im Listing festgehalten. Um den Platz in diesem Buch möglichst ökonomisch zu nutzen, wurden auf jede Seite drei Spalten gesetzt.

①

```
JUNIOR

1A7E
1A7E 04 CF.
1A7F 00 14.
1A80 80 201
0201 3C A0.
0202 3D 1.
0203 3C A9,
WHAT?
203
0203 3C A9.
0204 3C .
0205 3C 48.
0206 3C 8C.
0207 3C .
0208 3C 2.
0209 3C 68.
020A 3C 48.
020B 3C 18.
020C 3C 6D.
020D 3C .
020E 35 2.
020F 3C 48.
0210 2C C8.
0211 3C C0.
0212 3C B.
0213 3C D0.
0214 3C F1.
0215 3C .
0216 36 -
0215 00 -
0214 F1 -
0213 D0 -
0212 0B -
0211 C0 -
0210 C8 -
020F 48 -
020E 02 -
020D 00 -
020C 6D -
020B 18 -
```

②

```
020A 48 -
0209 68 -
0208 02 -
0207 00 -
0206 8C -
0205 48 -
0204 00 -
0203 A9 -
0202 01 -
0201 A0 -
0200 3C +
0201 A0 L 1
ACC: C3
Y  : C3
X  : C3
PC : E6C3
SP : 01FF
PR : 00000100
     NV BDIZC 200
0200 3C P
E6C3 E6 201
0201 A0 R 1
0203 A9 R 2
0205 48 R 3
0206 8C R 4
0209 68 R 5
020A 48 R 6
020B 18 R 7
020C 6D R 8
020F 48 R 9
0210 C8 R 10
0211 C0 R 11
0213 D0 L 12
ACC: 01
Y  : 02
X  : C3
PC : 0213
SP : 01FD
PR : 10100100
     NV BDIZC 200
0200 01 P
0213 D0 R 12
```

③

```
0206 8C R 13
0209 68 R 14
020A 48 R 15
020B 18 R 16
020C 6D R 17
020F 48 R 18
0210 C8 R 19
0211 C0 R 20
0213 D0 R 21
0206 8C R 22
0209 68 L 23
ACC: 03
Y  : 03
X  : C3
PC : 0209
SP : 01FC
PR : 10100100
     NV BDIZC 200
0200 03 P
0209 68 R 23
020A 48 R 24
020B 18 R 25
020C 6D R 26
020F 48 R 27
0210 C8 R 28
0211 C0 R 29
0213 D0 L 30
ACC: 06
Y  : 04
X  : C3
PC : 0213
SP : 01FB
PR : 10100100
     NV BDIZC 200
0200 03 P
0213 D0 R 30
0206 8C R 31
0209 68 R 32
020A 48 R 33
020B 18 R 34
020C 6D R 35
020F 48 R 36
```

④

```
0210 C8 R  37
0211 C0 R  38
0213 D0 L  39
ACC: 0A
Y  : 05
X  : C3
PC : 0213
SP : 01FA
PR : 10100100
     NV BDIZC 2000
2000 3C 200
0200 04 P
0213 D0 R  39
0206 8C R  40
0209 68 R  41
020A 48 R  42
020B 18 R  43
020C 6D R  44
020F 48 R  45
0210 C8 R  46
0211 C0 R  47
0213 D0 L  48
ACC: 0F
Y  : 06
X  : C3
PC : 0213
SP : 01F9
PR : 10100100
     NV BDIZC 200
0200 05 P
0213 D0 R  48
0206 8C R  49
0209 68 R  50
020A 48 R  51
020B 18 R  52
020C 6D R  53
020F 48 R  54
0210 C8 R  55
0211 C0 R  56
0213 D0 L  57
ACC: 15
Y  : 07
```

⑤

```
X  : C3
PC : 0213
SP : 01F8
PR : 10100100
     NV BDIZC 200
0200 06 P
0213 D0 R  57
0206 8C R  58
0209 68 R  59
020A 48 R  60
020B 18 R  61
020C 6D R  62
020F 48 R  63
0210 C8 R  64
0211 C0 R  65
0213 D0 L  66
ACC: 1C
Y  : 08
X  : C3
PC : 0213
SP : 01F7
PR : 10100100
     NV BDIZC 200
0200 07 P
0213 D0 R  66
0206 8C R  67
0209 68 R  68
020A 48 R  69
020B 18 R  70
020C 6D R  71
020F 48 R  72
0210 C8 R  73
0211 C0 R  74
0213 D0 L  75
ACC: 24
Y  : 09
X  : C3
PC : 0213
SP : 01F6
PR : 10100100
     NV BDIZC 200
0200 08 P
```

⑥

```
0213 D0 R  75
0206 8C R  76
0209 68 R  77
020A 48 R  78
020B 18 R  79
020C 6D R  80
020F 48 R  81
0210 C8 R  82
0211 C0 R  83
0213 D0 L  84
ACC: 2D
Y  : 0A
X  : C3
PC : 0213
SP : 01F5
PR : 10100100
     NV BDIZC 20
0200 09 P
0213 D0 R  84
0206 8C R  85
0209 68 R  86
020A 48 R  87
020B 18 R  88
020C 6D R  89
020F 48 R  90
0210 C8 R  91
0211 C0 R  92
0213 D0 L  93
ACC: 37
Y  : 0B
X  : C3
PC : 0213
SP : 01F4
PR : 00100111
     NV BDIZC 20
0200 0A P
0213 D0 R  93
0215 00 R  94
0217 3C R  95
021A 3C R
021D 3C R
0220 47 R
```

⑦

```
0222 C3 L
ACC: 56
Y  : 0B
X  : C3
PC : 0222
SP : 01F4
PR : 00100101
     NV BDIZC 1FF
01FF 00 -
01FE 01 -
01FD 03 -
01FC 06 -
01FB 0A -
01FA 0F -
01F9 15 -
01F8 1C -
01F7 24 -
01F6 2D -
01F5 37 -
01F4 10 -
01F3 89 -
01F2 12 -
01F1 07 1A7A
1A7A CF .
1A7B 14 1A.
1A7C 5F +
1A7D 10 +
1A7E CF .
1A7F 14 1A.
1A80 80 1A00
1A00 00 D8.
1A01 00 4C.
1A02 00 CF.
1A03 00 14.
1A04 00 217
0217 3C F8.
0218 2E 4C.
0219 3C FF.
021A 3C 1.
021B 3C 217
0217 F8 R 95
0218 4C L 96
```

⑧

```
ACC: 56
Y  : 0B
X  : C3
PC : 0218
SP : 01F4
PR : 00101101
     NV BDIZC P
0218 4C R 96
01FF 00 R 97
0201 A0 L 1
ACC: 56
Y  : 0B
X  : C3
PC : 0201
SP : 01F4
PR : 00111101
     NV BDIZC 200
0200 0A P
0201 A0 R 1
0203 A9 R 2
0205 48 R 3
0206 8C R 4
0209 68 R 5
020A 48 R 6
020B 18 R 7
020C 6D R 8
020F 48 R 9
0210 C8 R 10
0211 C0 R 11
0213 D0 L 12
ACC: 01
Y  : 02
X  : C3
PC : 0213
SP : 01F2
PR : 10101100
     NV BDIZC 200
0200 01 P
0213 D0 R 12
0206 8C R 13
0209 68 R 14
020A 48 R 15
```

⑨

```
020B 18 R 16
020C 6D R 17
020F 48 R 18
0210 C8 R 19
0211 C0 R 20
0213 D0 L 21
ACC: 03
Y  : 03
X  : C3
PC : 0213
SP : 01F1
PR : 10101100
     NV BDIZC 200
0200 02 P
0213 D0 R 21
0206 8C R 22
0209 68 R 23
020A 48 R 24
020B 18 R 25
020C 6D R 26
020F 48 R 27
0210 C8 R 28
0211 C0 R 29
0213 D0 L 30
ACC: 06
Y  : 04
X  : C3
PC : 0213
SP : 01F0
PR : 10101100
     NV BDIZC 200
0200 03 P
0213 D0 R 30
0206 8C R 31
0209 68 R 32
020A 48 R 33
020B 18 R 34
020C 6D R 35
020F 48 R 36
0210 C8 R 37
0211 C0 R 38
0213 D0 L 39
```

⑩

```
ACC:  10
Y   :  05
X   :  C3
PC  :  0213
SP  :  01EF
PR  :  10101100
       NV BDIZC 200
0200  04 P
0213  D0 R  39
0206  8C R  40
0209  68 R  41
020A  48 R  42
020B  18 R  43
020C  6D R  44
020F  48 R  45
0210  C8 R  46
0211  C0 R  47
0213  D0 L  48
ACC:  15
Y   :  06
X   :  C3
PC  :  0213
SP  :  01EE
PR  :  10101100
       NV BDIZC 200
0200  05 P
0213  D0 R  48
0206  8C R  49
0209  68 R  50
020A  48 R  51
020B  18 R  52
020C  6D R  53
020F  48 R  54
0210  C8 R  55
0211  C0 R  56
0213  D0 L  57
ACC:  21
Y   :  07
X   :  C3
PC  :  0213
SP  :  01ED
```

⑪

```
PR  :  10101100
       NV BDIZC 200
0200  06 P
0213  D0 R  57
0206  8C R  58
0209  68 R  59
020A  48 R  60
020B  18 R  61
020C  6D R  62
020F  48 R  63
0210  C8 R  64
0211  C0 R  65
0213  D0 L  66
ACC:  28
Y   :  08
X   :  C3
PC  :  0213
SP  :  01EC
PR  :  10101100
       NV BDIZC 200
0200  07 P
0213  D0 R  66
0206  8C R  67
0209  68 R  68
020A  48 R  69
020B  18 R  70
020C  6D R  71
020F  48 R  72
0210  C8 R  73
0211  C0 R  74
0213  D0 L  75
ACC:  36
Y   :  09
X   :  C3
PC  :  0213
SP  :  01EB
PR  :  10101100
       NV BDIZC 200
0200  08 P
0213  D0 R  75
0206  8C R  76
0209  68 R  77
```

⑫

```
020A  48 R  78
020B  18 R  79
020C  6D R  80
020F  48 R  81
0210  C8 R  82
0211  C0 R  83
0213  D0 R  84
0206  8C L  85
ACC:  45
Y   :  0A
X   :  C3
PC  :  0206
SP  :  01EA
PR  :  10101100
       NV BDIZC 20
0200  09 P
0206  8C R  85
0209  68 R  86
020A  48 R  87
020B  18 R  88
020C  6D R  89
020F  48 R  90
0210  C8 R  91
0211  C0 R  92
0213  D0 L  93
ACC:  55
Y   :  0B
X   :  C3
PC  :  0213
SP  :  01E9
PR  :  00101111
       NV BDIZC 20
0200  0A P
0213  D0 R  93
0215  00 L  94
ACC:  55
Y   :  0B
X   :  C3
PC  :  0215
SP  :  01E9
PR  :  00101111
       NV BDIZC 20
```

```
        ⑬
0200 0A 1FF
01FF 00 -
01FE 01 -
01FD 03 -
01FC 06 -
01FB 0A -
01FA 0F -
01F9 15 -
01F8 1C -
01F7 24 -
01F6 2D -
01F5 37 -
01F4 00 -
01F3 01 -
01F2 03 -
01F1 06 -
01F0 10 -
01EF 15 -
01EE 21 -
01ED 28 -
01EC 36 -
01EB 45 -
01EA 55 -
01E9 10
```

Programm NUMBERS dezimal

Wir gehen nun noch einen Schritt weiter und lassen die NUMBERS-Zahlenreihe vom Computer dezimal berechnen. Als Ergebnis müssen wir natürlich genau die gleichen Zahlen**werte** erhalten; der Computer rechnet jedoch im vertrauten Dezimalsystem und druckt auch das Ergebnis als Reihe von dezimalen Zahlen aus. Während zum hexadezimalen Zahlensystem die Ziffern Ø . . . 9 und A . . . F gehören, kommen im dezimalen Zahlensystem bekanntlich nur die Ziffern Ø . . . 9 vor.
Im Anhang 2 am Schluß dieses Buchs ist beschrieben, daß dann Komplikationen entstehen, wenn man über den SAVE-Eingang des Standard-Monitors zu diesem zurückkehren will und vorher dezimal gerechtnet hat (D-Flag wurde gesetzt). Ähnliche Schwierigkeiten tauchen auf, wenn die D-Flag beim Rücksprung zum STEP-Eingang von PM (Adresse 14CF) gesetzt ist. An verschiedenen Stellen von PM stehen die Befehle ADC und SBC; werden diese Befehle dezimal ausgeführt, dann kann PM nicht einwandfrei arbeiten. Außerdem werden Daten mit den Ziffern A . . . F falsch interpretiert, da diese Ziffern nur im hexadezimalen Zahlensystem vorkommen. In Junior-Computer-Buch 4 wird die PM-Software Byte für

Byte erläutert, so daß an dieser Stelle nicht weiter darauf eingegangen werden soll.
Ebenso wie der Standard-Monitor durch einige im PIA-RAM stehende Bytes ergänzt wurde (siehe Anhang 2), kann man auch bei PM die binäre Arbeitsweise wiederherstellen. Vor die STEP-Routine wird die im PIA-RAM stehende Routine BINAR gesetzt: siehe Bild 16b. Der NMI-Sprungvektor muß auf BINAR (Adresse 1AØØ) gerichtet sein, wenn man Anwenderprogramme schrittweise durchlaufen oder durch Drücken von Taste ST nach PM zurückkehren will. Kommen im Anwenderprogramm BRK-Befehle vor, so muß ferner auch der IRQ-Sprungvektor auf BINAR gerichet sein. Unbedingt erforderlich ist der Umweg über BINAR zu STEP in dem Fall, daß die D-Flag logisch 1 ist (dezimaler Rechenmodus). Wenn vorher binär gerechnet wurde (D-Flag Ø), dann ist dieser Umweg zwar nicht notwendig, aber auch nicht schädlich.

Noch etwas anderes: In Kapitel 10 wurde beschrieben, wie die Hardware der Junior-Computer-Basisplatine zu ergänzen ist, um das schrittweise Abarbeiten der Monitor-Programme (Standard und PM) zu verhindern. Da der Aufdruck "K4, K7" auf der Aufsteckplatine EPS 81033-3 Verwirrung stiften kann, geben wir hier zur Sicherheit noch einmal präzise Anweisungen für die Verlegung der Leitungen:

1. K7 muß stets angeschlossen sein, auch wenn die Vektoren RES, NMI und IRQ selbst im Buskartenspeicher stehen! Der Grund dafür ist folgender: Im Standard-Monitor-EPROM stehen zwei JMP-IND-Instruktionen, die ausgeführt werden müssen, wenn ein NMI oder IRQ tatsächlich erledigt werden soll (im Step-Modus ist dies unumgänglich; die effektiven NMI- und IRQ-Sprungvektoren stehen im PIA-RAM auf Seite 1A).
2. Wenn man ohne PM, also mit dem Standard-Monitor dezimal rechnen will, ist K6 der zweite Anschluß. Alles weitere siehe Anhang 2 am Schluß dieses Buchs.
3. Der zweite Anschluß ist K4, wenn man wie beim Programm NUMBERS (Tabelle 2, erste Hälfte) schrittweise mit PM arbeiten will.

NUMBERS soll nun jedoch dezimal abgearbeitet **und** schrittweise durchlaufen werden. Angesichts der gewählten Lösung (siehe BINAR, Bild 16b) ist als dritte Leitung zur Aufsteckplatine die Verbindung mit K6 notwendig. Wie sie angeschlossen wird, geht aus Bild 17 hervor. Zugegeben: Die Diodenschaltung ist nicht besonders elegant, sie erfüllt jedoch einwandfrei ihren Zweck. Damit sind alle hardwaremäßigen Voraussetzungen geschaffen, um NUMERS im dezimalen Modus schrittweise durchlaufen zu können.
Wie gehen wir von der Seite der Software aus zu Werk? Die Antwort ergibt sich aus folgenden Überlegungen:

- Nach Ausführen der binären Version von NUMBERS steht der Programmzähler wegen des BRK-Bremsweges auf Adresse Ø217 mit dem Label DECAR. Daß tatsächlich binär gearbeitet wurde, beweist übrigens das Ergebnis: die im Stack stehende Zahlenreihe. Nach dem Start von PM wird nämlich ebenso wie beim RESET-Vorprogramm des Standard-Monitors der Befehl CLD ausgeführt.
- Bei Adresse Ø217 beginnt das Programm aus Bild 16a. Dieses Programm kann der Prozessor zusammenhängend abarbeiten; es braucht nicht schrittweise durchlaufen zu werden. Erst wird die D-Flag gesetzt (SED),

dann folgt ein Sprung zur niedrigsten Stack-Adresse, die bei dieser Gelegenheit das Label STACK erhielt.

- Im Speicherplatz Ø1FF stehen die Daten ØØ; dies ist die erste Zahl der NUMBERS-Zahlenreihe. Die **ØØ dient jetzt als Opcode der Instruktion BRK!** Nach Ausführen des BRK steht Programmzähler PC auf Adresse Ø2Ø1, der Startadresse von NUMBERS in der binären und auch in der dezimalen Version.
- Der Sprung von DECAR nach NUMBERS infolge des BRK bei Adresse Ø1FF (siehe auch Bild 16b) hat zur Folge, daß das unveränderte Programm NUMBERS mit geänderter Startadresse dezimal statt hexadezimal ausgeführt wird.

Und so setzen wir vorstehende Überlegungen in die Praxis um:

- Ausgangspunkt ist das Ende des binären Durchlaufs von NUMBERS; der Programmzähler steht hier auf Ø217 (läßt sich durch Drücken von P kontrollieren). Ausgedruckt wurden an dieser Stelle die Inhalte der Prozessor-Register und des Speicherplatzes TEMPY sowie die im Stack stehende Zahlenreihe.
- Jetzt werden die Daten aus den Bildern 16a und 16b eingetippt. Nicht vergessen: die NMI- und IRQ-Sprungvektoren sind neu zu spezifizieren!
- Der STEP-Schalter wird auf OFF gesetzt. Das ist zwar nicht zwingend notwendig, doch so erreichen wir die Startadresse von NUMBERS am schnellsten. In Tabelle 2 wurde schrittweise gearbeitet; Pausen entstanden bei den Zwischenstationen 96 und 97 aus Bild 16c.
- Um zur Startadresse DECAR zurückzukehren, wird Taste P gedrückt.
- Wir drücken nun Taste R: Das Programm läuft; über PM folgt die Rückmeldung mit der Startadresse von NUMBERS.
- Vor dem Start von NUMBERS lassen wir die Inhalte der Prozessorregister und des Speicherplatzes TEMPY ausdrucken. Dazu drücken wir die Tasten L und 2ØØ (SP).
- Wir stellen den STEP-Schalter auf ON und drücken dann die Taste P (Sprung zur Startadresse) sowie die Taste R (Ausführen der ersten Instruktion).

Der weitere Ablauf ist der gleiche wie beim binären Abarbeiten von NUMBERS; er läßt sich an Hand von Tabelle 2 leicht verfolgen.

Noch eine Anmerkung: Wir sind beim Programmbeispiel NUMBERS so vorgegangen, daß das Verfahren der schrittweisen Programmausführung möglichst deutlich zu Tage trat. Es war beispielsweise nicht unbedingt notwendig, über das Programm DECAR (Bild 16a) zur dezimalen Rechenweise überzugehen. Auch hätten wir bereits beim binären Durchlauf von NUMBERS über den BINAR-Umweg (Bild 16b) nach PM zurückspringen können. Mit anderen Worten: Die einzige Änderung, die nach dem binären Durchlauf von NUMBERS vorgenommen werden mußte, betrifft den Übergang auf den dezimalen Rechenmodus des Prozessors. Diese Änderung kann nicht nur ein Anwenderprogramm mit dem Befehl SED bewirken, sondern sie kann auch unmittelbar durch PM zustande kommen: Man lade den Speicherplatz PREG (Adresse ØØF1) so, daß die D-Flag auf 1 gesetzt wird, während alle übrigen Flags unverändert bleiben.

Inzwischen ist sicherlich deutlich geworden, daß das schrittweise Durchlaufen von Programmen mehr Möglichkeiten bietet als nur das Feststellen

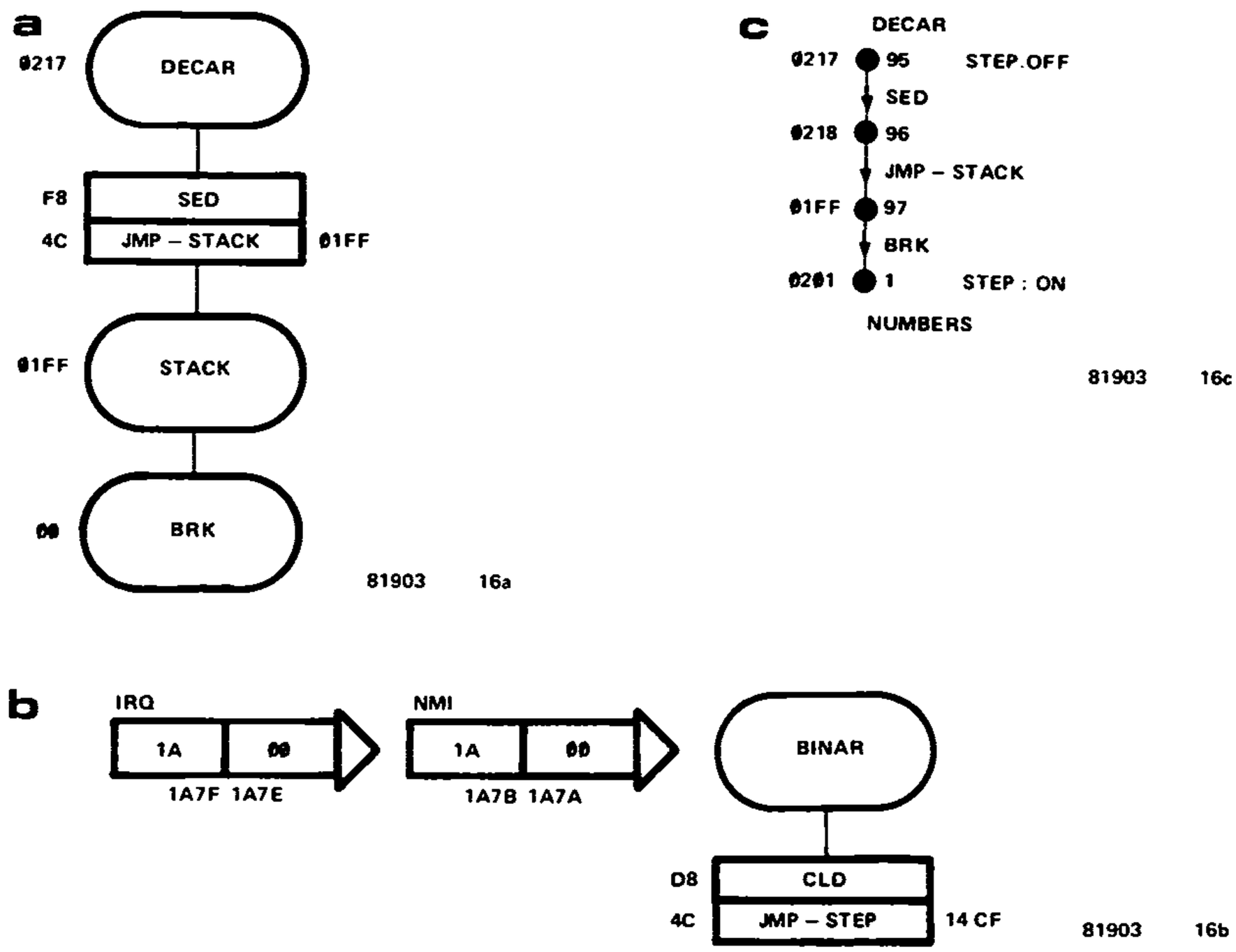


Bild 16. Mit Hilfe des Programms DECAR (a) kann NUMBERS nach dem binären Durchlauf noch einmal im dezimalen Modus durchlaufen werden. PM wird dazu durch einige im PIA-RAM untergebrachte Bytes ergänzt (b). Auch die IRQ- und NMI-Sprungvektoren sind anzupassen. Im Schienen-Modell aus Bild 15 läuft dies auf eine Erweiterung des Schienennetzes hinaus (c).

der Prozessor-Registerinhalte in unterschiedlichen Phasen des Programmablaufs. Man kann darüber hinaus den Programmablauf durch Modifikation von Daten gezielt beeinflussen. Ferner wurde deutlich, daß sich das "Durchschreiten" eines Programms mit PM wesentlich bequemer gestaltet als mit dem Standard-Monitor.

PM: Pro Memoria

PM-Merkzettel

Ändern des NMI-Sprungvektors. Nach dem Start von PM wird der NMI-Sprungvektor (Adressen 1A7A und 1A7B) automatisch auf Adresse 14CF gerichtet. Dies ist die Startadresse des STEP-Vorprogramms, das die Inhalte der Prozessor-Register notiert und den Stand des Programmzählers als aktuelle Arbeitsadresse ausdruckt. Man kann jedoch den NMI-Sprungvektor auch auf eine andere Adresse richten, zum Beispiel, wenn Taste ST auf der Basiskarte eine spezielle Funktion erhalten soll. Dabei ist auch an das mögliche "Überlaufen" des Stack zu denken; jede Betätigung von ST füllt im Stack drei weitere Speicherplätze.

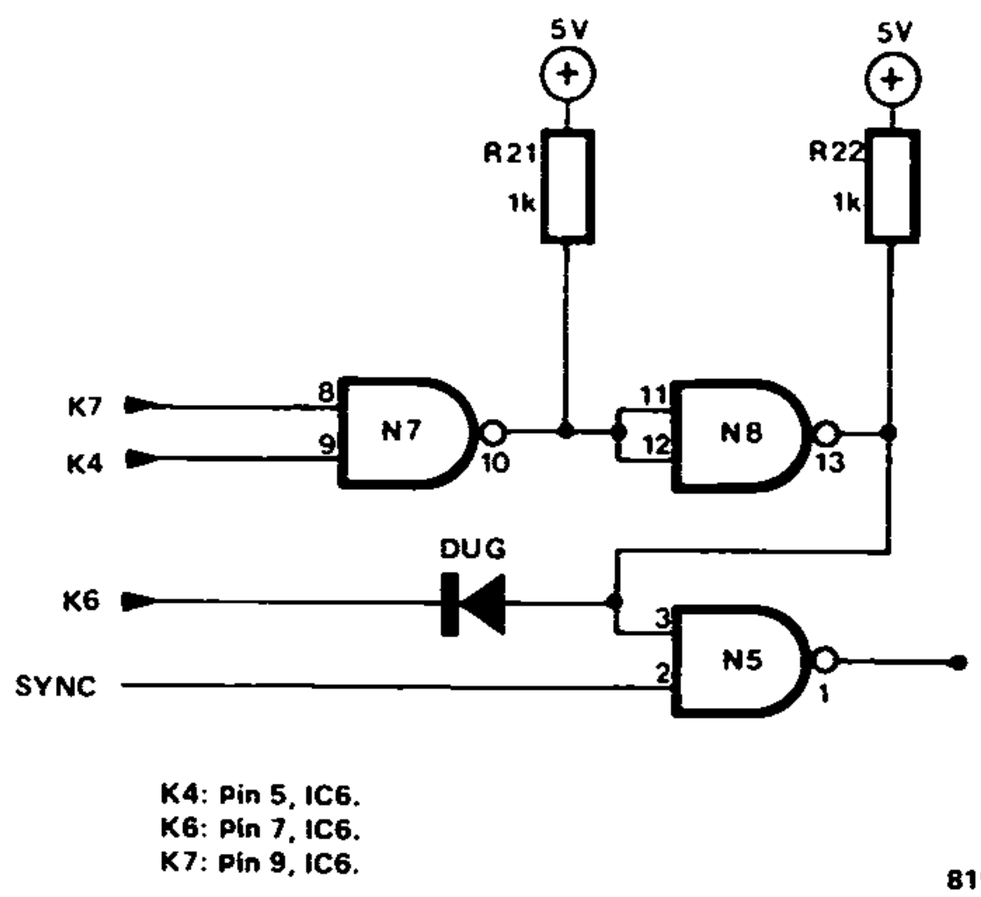


Bild 17. Wenn man unter Regie vom PM dezimal rechnen will, dann muß auch Anschluß K6 des Adreßdekoders auf der Basisplatine mit der Aufsteckplatine verbunden werden. Dazu ist eine Germaniumdiode (DUG) notwendig.

Wirksam geändert werden kann der NMI-Sprungvektor nur während des Verbleibs in PM. Verläßt man PM und kehrt später wieder nach PM zurück, dann werden die NMI-Sprungvektor-Adressen 1A7A und 1A7B mit 14CF überschrieben. Der IRQ-Sprungvektor kann dagegen zu einem beliebigen Zeitpunkt spezifiziert werden.

Eine Warnung. Wenn man von den Tastenfunktionen S oder G Gebrauch macht, also eine Subroutine von TM benutzt, und gleichzeitig der STEP-Schalter auf ON steht, können Datensendungen weder auf Band geschrieben noch vom Band gelesen werden. Die Rückmeldung "READY" läßt dann unendlich lange auf sich warten. TM ist nämlich in einem EPROM gespeichert, das von den Signalen K2 und K3 selektiert wird (siehe Kapitel 10). Sind aber diese Selektionssignale aktiv, dann ist das schrittweise Abarbeiten des dort stehenden Programms **nicht** blockiert! Der STEP-Schalter muß deshalb beim Gebrauch der Tastenfunktionen S und G auf OFF stehen.

Fehlermeldungen

Nur ein Teil der Tasten auf der ASCII-Tastatur hat bei PM eine Funktion (siehe Bild 12); mit den ASCII-Zeichen der übrigen Tasten kann PM nichts anfangen Ferner ist PM ratlos, wenn bei den Tastenfunktionen M, S und G die Parameter nicht in vorgeschriebener Weise eingegeben werden. Wir drücken beispielsweise die Tasten
S11, 2ØØ, R
und stellen fest, daß keine Datensendung auf Band geschrieben wird und PM auch nicht auf des Kommando R reagiert. Statt dessen wird
JUNIOR
ausgedruckt, gefolgt von einer Leerzeile.
Es gibt noch eine Reihe weiterer Anlässe, die zu Fehlermeldungen führen.

a. Rückmeldung "WHAT?"

Diese Rückmeldung erscheint, in bestimmten Fällen gefolgt von einer Leerzeile und "JUNIOR", als Reaktion auf das Drücken einer Taste, die nicht numerisch ist und auch keine relevante oder überhaupt keine PM-Tastenfunktion verkörpert. Wird eine Taste mit parametrisierter Funktion (M, S oder G) gedrückt, dann folgt auf die Frage "WHAT?" stets die Meldung "JUNIOR" sowie eine weitere Leerzeile. Ein Beispiel:

0407 3F,
WHAT?

Wir wollten in Speicherplatz 0407 die Daten 3F setzen, haben jedoch an Stelle des Punktes ein Komma eingetippt. Das Komma hat aber nur bei den Tastenfunktionen M und S eine Bedeutung.

b. Rückmeldung "JUNIOR"

Die Rückmeldung JUNIOR, der wir schon verschiedentlich begegnet sind, erscheint in folgenden Fällen:

1. nach dem vollständigen Start von PM (erster und zweiter Start);
2. nach Drücken der BRK-Taste, während der Junior-Computer ein oder mehrere ASCII-Zeichen sendet. Voraussetzung ist, daß der BRK-Sprungvektor wie folgt spezifiziert ist: Adresse 1A7C → Daten 5F, Adresse 1A7D → Daten 10. Beim Start von PM wird der BRK-Sprungvektor automatisch auf Adresse 105F gerichtet; er kann jedoch später vom Programmierer geändert werden;
3. nach der vollständigen Eingabe des Befehls zum Ausdrucken eines Hexdump, also nach Drücken von M und Eingabe der Parameter.

In den drei vorstehenden Fällen ist "JUNIOR" als Rückmeldung und nicht als Fehleranzeige zu verstehen. Eingabefehler werden jedoch ebenfalls durch "JUNIOR" signalisiert, wenn

4. nach Drücken von M zwar die Parameter formal richtig eingegeben wurden, die Endadresse des auszudruckenden Datenblocks aber niedriger als die Beginnadresse ist;
5. nach Drücken von S als Programmnummer 00 oder FF eingegeben wird. Diese Programmnummern sind für das Schreiben auf Band nicht zugelassen, sie haben beim Lesen vom Band Sonderfunktionen;
6. bei den Tastenfunktionen M, S oder G die Parameter falsch eingegeben werden. In bestimmten Phasen der Eingabe spielen ausschließlich die numerischen Tasten, die Kommataste (außer bei G) und die Taste CR eine Rolle. Werden andere Tasten gedrückt, so folgt eine Fehlermeldung. Das Gleiche gilt, wenn beispielsweise Taste CR vor Eingabe des zweiten Parameters gedrückt wird. Mit anderen Worten: Jedes Abweichen von den Eingabeschemen

M → **HEXDUMP:** (numerische Daten), (numerische Daten)CR
S (numerische Daten), (numerische Daten), (numerische Daten)CR
G (numerische Daten ≠ FF)CR
GFF → **SA:** (numerische Daten)CR

führt zu einer Fehlermeldung. Auch das unplanmäßige Drücken der Zwischenraumtaste SP, zum Beispiel um ein optisch anderes Textbild zu erhalten, wertet PM als Fehler. In einigen Fällen geht einer Fehlermeldung während der Vorbereitungsphase einer parametrisierten Tastenfunktion

die Meldung "WHAT?" voraus. Die Beschreibung der PM-Software in Buch 4 geht hierauf näher ein.
Was unternimmt man nach einer Fehlermeldung? Handelt es sich um eine parametrisierte Tastenfunktion (M, S oder G), dann wird M, S oder G noch einmal gedrückt und die Eingabe wiederholt. In den anderen Fällen läuft die Korrektur meistens darauf hinaus, daß die Arbeitsadresse noch einmal eingegeben werden muß. Man drückt die betreffenden numerischen Tasten und anschließend die Zwischenraumtaste SP.

Noch eine Anmerkung zu S und G: Wenn S oder G gedrückt wurde, die Parameter eingegeben sind und CR gedrückt ist, wird auch das grafische Kommando CR (zurück zum Zeilenanfang) ausgeführt. Der Cursor auf dem Bildschirm steht jetzt am Anfang der gerade geschriebenen Zeile, so daß das dort stehende Zeichen (Buchstabe S bzw. G) bis zum Erscheinen der Rückmeldung READY periodisch unsichtbar ist.

Zugabe

Hinter den Kulissen von PM

Welche Subroutinen gehören zu PM und wie lassen sie sich für Anwenderprogramme nutzen? Eine umfassende Antwort auf diese Frage gibt Junior-Computer-Buch 4. Zum Abschluß dieses Kapitels wollen wir jedoch einen ersten Blick hinter die Kulissen von PM werfen und an vier Beispielen die Verwendungsmöglichkeiten der PM-Subroutinen demonstrieren.

1. Ausdrucken von Daten

Wie wir inzwischen wissen, druckt PM auf Tastendruck (Taste M) ein Hexdump aus. Ebenso kann PM Daten von Anwenderprogrammen, zum Beispiel die vom Programm NUMBERS (Bild 13) produzierten Zahlenreihen ausdrucken. Hierzu dient das Programm PRNUMB in Bild 18, das drei verschiedene Subroutinen von PM benutzt. Diese Subroutinen heißen

- CRLF: Beginne am Anfang einer neuen Zeile (11E8),
- PRSP: Schiebe einen Zwischenraum ein (11F3) und
- PRBYT: Sende den Inhalt des Akkus in Form von zwei ASCII-BYTES zur Peripherie (128F).

Wie ist PRNUMB aufgebaut? Der Anfang stimmt mit NUMBERS (siehe Bild 13) überein. Die "BNE-Weiche" im Programmteil nach dem Label ADD steht auf Sprung, solange noch nicht sämtliche elf Zahlen der NUMBERS-Zahlenreihe berechnet und in den Stack gesetzt sind. Ist die Zahlenreihe vollständig, so wird der Zustand der D-Flag überprüft. Dazu wird durch die Befehle PHP und PLA der Inhalt des P-Registers in den Akku kopiert. Die Instruktion AND #Ø8 setzt alle Bits des Akkus (nicht des P-Registers!) mit Ausnahme des zur D-Flag gehörenden Bit auf Null. Dieses Bit entscheidet, ob der Programmteil nach ADD ein einziges Mal (hexadezimaler Rechenmodus) oder bereits ein zweites Mal (dezimaler Rechenmodus) durchlaufen wurde. Wenn D logisch Ø ist, folgt der Wechsel zum dezimalen Rechenmodus (SED) und der Sprung zurück nach PRNUMB.

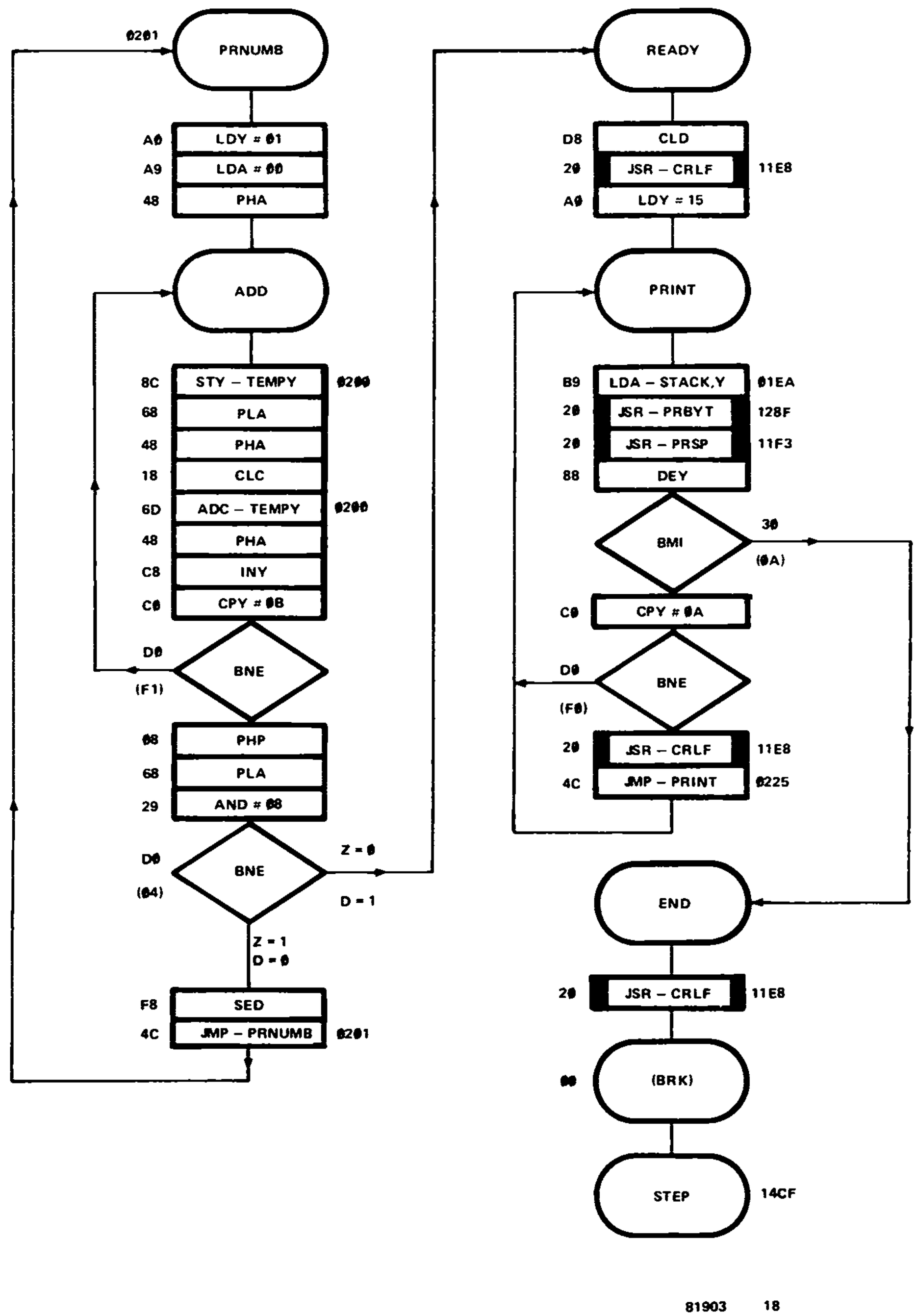


Bild 18. Die Zahlenreihe NUMBERS (siehe Bild 13) kann mit dem Programm PRNUMB binär und dezimal berechnet werden. Der Inhalt des Stack (siehe Bild 14) wird als binäre (hexadezimale) und dezimale Zahlenreihe ausgedruckt.

Weshalb wird die erste Berechnung der Zahlenreihe binär ausgeführt? Weil natürlich das Programm zuerst mit dem Standard-Monitor oder mit PM (letzteres verdient sicherlich den Vorrang) eingetippt oder eingelesen werden mußte. Nach dem Start des Standard-Monitors oder dem Start von PM arbeitet der Prozessor automatisch im binären (hexadezimalen) Rechenmodus.
Sobald die Prüfung der D-Flag ergibt, daß diese als Zeichen für das vollständige Berechnen beider Zahlenreihen log. 1 geworden ist, folgt der Sprung nach READY. Dieser zweite Teil von PRNUMB sorgt für das geordnete Ausdrucken der Zahlen. Hier taucht die Frage auf, ob es nicht möglich war, die Zahlen einzeln innerhalb der ADD-Schleife auszudrucken. Die Antwort lautet: Ja, während des binären Rechenmodus ist dies möglich, beim dezimalen Rechenmodus jedoch nicht. Dies liegt daran, daß die PM-Subroutinen mit großer Wahrscheinlichkeit nicht einwandfrei laufen, wenn sie bei gesetzter D-Flag über einen JSR angesprungen werden. Der Sprung nach PM (Step-Eingang) über den BINAR-Umweg verläuft bei D=1 zwar problemlos, läßt man aber den Umweg weg und tritt bei D=1 über eine der Subroutinen in PM ein, dann ergeben sich ernsthafte Schwierigkeiten. Deshalb setzt der CLD die D-Flag unmittelbar nach dem Label READY auf Ø. Der Sprung zur Subroutine CRLF sorgt dafür, daß anschließend eine neue Zeile begonnen wird.
Die beiden Zahlenreihen sollen in übersichtlicher Form untereinander ausgedruckt werden: zuerst die binäre, darunter die dezimale Zahlenreihe. Dazu setzen wir den Wert 15 in das Y-Register und laden den Akku über die absolute, Y-indizierte Adressierung mit den Inhalten der Stack-Speicherplätze (Operandenadresse Ø1EA). Wie das Stack-Modell in Bild 14 zeigt, liegt der Speicherplatz Ø1FF um hexadezimal 15 (dezimal: 21) Plätze von Ø1EA entfernt; in Ø1FF befindet sich die erste Zahl der binären Zahlenreihe. Durch Dekrementieren des Y-Registers werden nacheinander alle Zahlen in den Akku gesetzt.
Auf das Laden des Akkus folgen die Subroutinen PRBYT (Akku-Inhalt ausdrucken) und PRSP (Zwischenraum einfügen). Der Inhalt des Y-Registers wird, wie schon erwähnt, um eins vermindert (DEY); anschliessend wird auf die gleiche Weise die nächste Zahl ausgedruckt. Ist der Y-Inhalt ØØ, dann ist die letzte Zahl an der Reihe. Beim nächsten DEY springt Y auf FF, so daß die N-Flag gesetzt wird und über den BMI der Sprung zum Label END folgt: Beginn einer neuen Zeile (CRLF) mit anschließendem BRK, also ein Sprung zum Step-Eingang von PM.
Wenn im Y-Register der Wert ØA steht, muß PRNUMB die restlichen Zahlen auf einer neuen Zeile ausdrucken. Aus Bild 14 geht hervor, daß der Wert ØA im Y-Register dem ersten Glied der dezimalen Zahlenreihe entspricht. Die Instruktionen CPY #ØA und BNE PRINT sorgen für den neuen Zeilenanfang (CRLF).
Nach vorstehenden Erläuterungen ist es nun an der Zeit, das Programm PRNUMB in den Computer einzugeben. Nachfolgend ist PRNUMB in der übersichtlichen Assembler-Schreibweise aufgelistet:

```
PRNUMB  0201  A0  01      LDY #01
        0203  A9  00      LDA #00
        0205  48          PHA
ADD     0206  8C  00  02  STY-TEMPY
```

```
        Ø2Ø9  68          PLA
        Ø2ØA  48          PHA
        Ø2ØB  18          CLC
        Ø2ØC  6D  ØØ  Ø2  ADC-TEMPY
        Ø2ØF  48          PHA
        Ø21Ø  C8          INY
        Ø211  CØ  ØB      CPY #ØB
        Ø213  DØ  F1      BNE ADD
        Ø215  Ø8          PHP
        Ø216  68          PLA
        Ø217  29  Ø8      AND #Ø8
        Ø219  DØ  Ø4      BNE READY
        Ø21B  F8          SED
        Ø21C  4C  Ø1  Ø2  JMP-PRNUMB
READY   Ø21F  D8          CLD
        Ø22Ø  2Ø  E8  11  JSR-CRLF
        Ø223  AØ  15      LDY #15
PRINT   Ø225  B9  EA  Ø1  LDA-(STACK,Y)
        Ø228  2Ø  8F  12  JSR-PRBYT
        Ø22B  2Ø  F3  11  JSR-PRSP
        Ø22E  88          DEY
        Ø22F  3Ø  ØA      BMI END
        Ø231  CØ  ØA      CPY #ØA
        Ø233  DØ  FØ      BNE PRINT
        Ø235  2Ø  E8  11  JSR-CRLF
        Ø238  4C  25  Ø2  JMP-PRINT
END     Ø23B  2Ø  E8  11  JSR-CRLF
        Ø23E  ØØ          BRK
        1A7E  CF          BRKL
        1A7F  14          BRKH
```

Nach vollendeter Tipparbeit starten wir das Programm PRNUMB:

2Ø1 (SP)
Ø2Ø1 AØ R
ØØ Ø1 Ø3 Ø6 ØA ØF 15 1C 24 2D 37
ØØ Ø1 Ø3 Ø6 1Ø 15 21 28 36 45 55
(Leerzeile)
Ø24Ø XX

Das Programm läuft einwandfrei; wir setzen es deshalb vor Ausschalten des Junior-Computers auf Band:

S1,2Ø1,23F (CR)
READY

Im Prinzip kann man auch Programme, die PM-Subroutinen enthalten, unter der Regie des Standard-Monitors ausführen lassen. Voraussetzung ist, daß der BRK-Sprungvektor (IRQ-Vektor) auf Adresse 1CØØ gerichtet wird (die Rückmeldung erscheint dann nicht auf dem Bildschirm, sondern auf den Siebensegmentdisplays), und daß PM nach Einschalten des Junior-Computers mindestens einmal gestartet wurde. Letzteres ist notwendig, weil bestimmte, zu messende und in RAM-Speicherplätze zu setzende Zeitwerte (Bitzeiten) für verschiedene PM-Subroutinen Bedeutung haben.

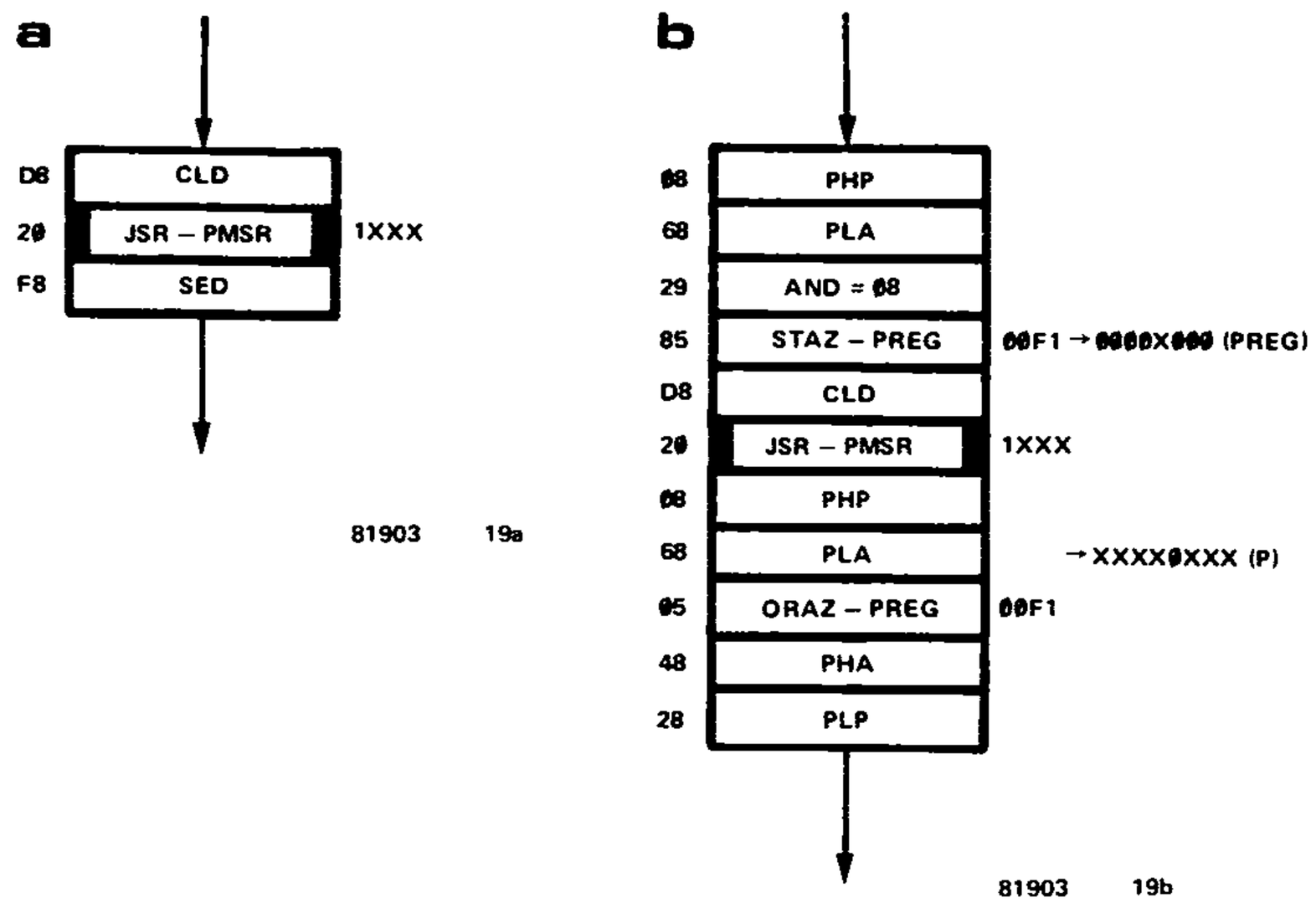


Bild 19. Die PM-Subroutinen dürfen nicht aufgerufen werden, solange der Prozessor im dezimalen Modus arbeitet. Rechnet das Anwenderprogramm ausschließlich dezimal, dann muß für die Dauer der PM-Subroutine auf den binären Modus umgeschaltet werden (a). Rechnet es abwechselnd binär und dezimal, dann ist der ursprüngliche Zustand der D-Flag nach dem Rücksprung aus PM wiederherzustellen (b). Das Durchlaufen von PM-Subroutinen ändert auch die übrigen Bit (Flags) des P-Registers.

Ein wichtiger Hinweis

Wie wir gesehen haben, stoßen wir beim Aufruf von PM-Subroutinen während des dezimalen Rechenmodus auf Schwierigkeiten. Diese werden vom Programm PRNUMB auf die beschriebene Weise umgangen. Eine andere Möglichkeit ist der Wechsel zum binären Rechenmodus vor Anspringen einer PM-Subroutine und die Rückkehr zum dezimalen Rechenmodus unmittelbar nach dem Rücksprung. Es können zwei unterschiedliche Situationen auftreten:

a. Der Prozessor arbeitet außerhalb von PM ausschließlich im dezimalen Rechenmodus. Hier ist die Lösung des Problems einfach: Vor den Sprung zu einer PM-Subroutine (Label PMSR) wird ein CLD gesetzt, nach dem Rücksprung folgt ein SED; siehe Bild 19a.

b. Der Prozessor rechnet binär und dezimal. In diesem Fall ist die Lösung etwas schwieriger; siehe Bild 19b: Mit den Befehlen PHP und PLA kopieren wir den Inhalt des P-Registers in den Akku und setzen durch AND #08 alle Akku-Bits mit Ausnahme des D-Flag-Bit auf 0. Das Ergebnis wird in Speicherplatz PREG (Adresse 00F1) gesetzt; dieser Puffer ist unbenutzt, da ja die PM-Subroutinen ebenso wie die Subroutinen des Standard-Monitors nicht schrittweise durchlaufen werden können. Dann folgen ein CLD sowie der Aufruf und die Erledigung der PM-Subroutine. Die meisten PM-Subroutinen setzen oder löschen bestimmte Flags; Einzelheiten werden ausführlich in Buch 4 behandelt. In unserem Fall müssen wir

nur den ursprünglichen Zustand der D-Flag wiederherstellen. Dazu kopieren wir mit den Befehlen PHP und PLA das P-Register in den Akku und lassen die Instruktion ORAZ-PREG folgen. Nach den Befehlen PHA und PLP ist der ursprüngliche Zustand der D-Flag im P-Register wieder vorhanden.

2. Grafiken

Die Subroutine PRCHA ist uns bereits begegnet: sie wird zweimal nacheinander innerhalb der Subroutine PRBYT aufgerufen, die wir im ersten Beispiel (PRNUMB; Bild 18) kennengelernt haben. Die Aufgabe von PRCHA ist das Senden und Ausdrucken des Akku-Inhalts. Wichtig ist hierbei die Tatsache, daß die Inhalte des X-Registers vor und nach PRCHA identisch sind, obwohl das X-Register während dieser Subroutine intensiv benutzt wird. Die Startadresse von PRCHA ist 1334.
Wir haben bereits numerische Daten ausdrucken lassen (erstes Beispiel), werden Texte auf den Bildschirm oder zu Papier bringen (später folgendes drittes Beispiel) und wollen uns nun zunächst mit einem grafischen Muster beschäftigen. Auch bei diesem zweiten Beispiel für die Nutzung von PM-Subroutinen gilt: Nicht der praktische Wert des Programms steht im Vordergrund, sondern die nähere Bekanntschaft mit der unerschöpflichen "Software-Trickkiste".
Auf den Bildschirm soll ein Schachbrettmuster, bestehend aus acht Reihen und acht Spalten, gesetzt werden. Jede Reihe und jede Spalte soll aus vier schwarzen und vier weißen Feldern in abwechselnder Reihenfolge zusammengesetzt sein. Wir vereinfachen das Schachbrett in der Weise, daß wir weiße Felder als Zwischenräume und schwarze Felder als Buchstaben "X" ausdrucken lassen. Das Schachbrett soll möglichst quadratische Form haben, so daß außer den Zwischenräumen, die die weißen Felder symbolisieren, in regelmäßigen Abständen weitere Zwischenräume eingefügt werden müssen.
Wie gehen wir ans Werk? Zuerst betrachten wir diese beiden Zahlenreihen:
ØØ Ø1 Ø2 Ø3 Ø4 Ø5 Ø6 Ø7 Ø8 Ø9 ØA ØB ØC ØD ØE ØF 1Ø
2Ø 2Ø 58 2Ø 2Ø 2Ø 58 2Ø 2Ø 2Ø 58 2Ø 2Ø 2Ø 58 ØD ØA
In der ersten Zeile stehen die Zahlen des Hexadezimalsystems in ihrer natürlichen Folge, darunter steht jeweils ein ASCII-Kode: 2Ø bedeutet "Zwischenraum", 58 ist der Kode für den Buchstaben X, ØD bedeutet "zurück zum Zeilenanfang" (CR) und ØA bedeutet "neue Zeile" (LF). Das Y-Register nimmt nacheinander die in der ersten Zeile stehenden Werte an, während die Daten der zweiten Reihe nacheinander mit Hilfe der absoluten, Y-indizierten Adressierung in den Akku gesetzt und anschließend von PRCHA ausgedruckt werden. Die Datenreihe, die einer Zeile des Schachbretts entspricht, wird insgesamt neun mal verarbeitet. Beim ersten Mal sorgt sie allerdings nur dafür, daß das Ausdrucken am Zeilenanfang beginnt.
Das X-Register hält den aktuellen Stand der Dinge fest. Sein Inhalt durchläuft nacheinander die Werte ØØ . . . Ø9; diese Zahlen haben folgende Bedeutung:
Bei X=ØØ ist Y=ØF: es wird eine neue Zeile begonnen. Bei X=Ø1, Ø3, Ø5 oder Ø7 ist Y=ØØ: es wird mit einem weißen Feld begonnen.

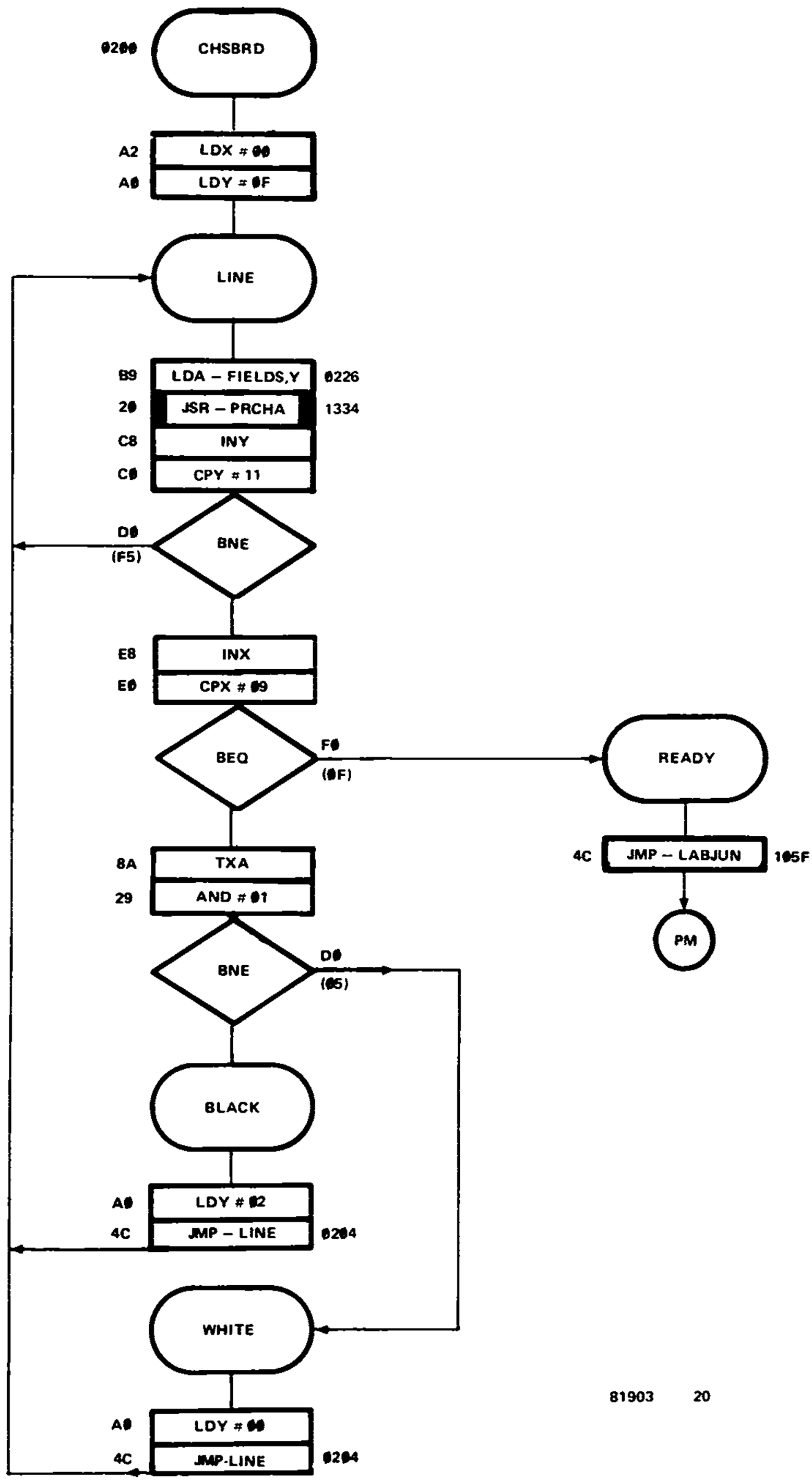


Bild 20. Das Programm CHSBRD setzt ein schachbrettartiges Muster auf den Bildschirm.

Bei X=Ø2, Ø4, Ø6 oder Ø8 ist Y=Ø2: es wird mit einem schwarzen Feld begonnen.
Ist X=Ø9, dann hat der Computer das Schachbrett vollendet: er fügt dann noch "JUNIOR" hinzu.
Damit ist das Schachbrettprogramm CHSBRD in Bild 20 schon fast erklärt. Ergänzend sei nur noch folgendes hinzugefügt: Die Programmschleife zwischen dem Label LINE und dem nächsten BNE wird während des Ausdruckens einer Schachbrettzeile durchlaufen. Tabelle FIELDS hat die Funktion der Datenzeile, die sich aus den ASCII-Kodes zusammensetzt. Aus einem der zu dieser Tabelle gehörenden Speicherplätzen wird der ASCII-Kode gelesen, der als Zeichen ausgedruckt werden soll. Ist das geschehen, dann wird der Inhalt des Y-Registers um eins erhöht. Bei Y=11 ist die Zeile vollständig; der Inhalt des X-Registers wird um eins erhöht und mit dem Wert Ø9 (= letzte Zeile) verglichen.
Wenn noch mindestens eine Schachbrettzeile fehlt, wird der Inhalt von X eingehender überprüft: Ist die dort stehende Zahl gerade oder ungerade? Muß also die nächste Zeile mit Y=Ø2 (Label BLACK; schwarzes Feld) oder mit Y=ØØ (Label WHITE; weißes Feld) beginnen? Um zu prüfen, ob die in X stehende Zahl gerade oder ungerade ist, wird X in den Akku kopiert (TXA); anschließend werden bis auf das ganz rechts stehende Bit (LSB) alle übrigen Bits durch AND # Ø1 auf Ø gesetzt. Bei einer ungeraden Zahl in X wird die Z-Flag Ø, bei einer geraden Zahl wird sie 1. Der Befehl BEQ stellt in Abhängigkeit von der Z-Flag die Weiche für den neuen Zeilenbeginn.
Steht das Schachbrett vollständig auf dem Schirm (X=Ø9), dann folgen ein Sprung zum Label READY in Bild 20 und anschließend ein Sprung zum Label LABJUN von PM. Der Programmteil nach LABJUN sorgt für die Rückmeldung "JUNIOR"; danach springt der Prozessor zur Warteschleife der Tastenabfrage von PM.
Nach der Theorie nun zur Praxis: Hier ist, wieder in Assembler-Schreibweise, das in den Computer einzugebende Programm aufgelistet:

```
CHSBRD   ØØØØ  A2  ØØ        LDX #ØØ
         ØØØ2  AØ  ØF        LDY #ØF
LINE     ØØØ4  B9  26  ØØ    LDA-FIELDS, Y
         ØØØ7  ØØ  34  13    JSR-PRCHA
         ØØØA  C8            INY
         ØØØB  CØ  11        CPY #11
         ØØØD  DØ  F5        BNE LINE
         ØØØF  E8            INX
         ØØ1Ø  EØ  Ø9        CPX #Ø9
         ØØ12  FØ  ØF        BEQ READY
         ØØ14  8A            TXA
         ØØ15  29  Ø1        AND #Ø1
         ØØ17  DØ  Ø5        BNE WHITE
BLACK    ØØ19  AØ  Ø2        LDY #Ø2
         ØØ1B  4C  Ø4  Ø2    JMP-LINE
WHITE    ØØ1E  AØ  ØØ        LDY #ØØ
         ØØ2Ø  4C  Ø4  Ø2    JMP-LINE
READY    ØØ23  4C  5F  1Ø    JMP-LABJUN
FIELDS   ØØ26  ØØ
```

```
0227  20
0228  58
0229  20
022A  20
022B  20
022C  58
022D  20
022E  20
022F  20
0230  58
0231  20
0232  20
0233  20
0234  58
0235  0D
0236  0A
```

Einmal Eintippen genügt. Deshalb starten wir zuerst noch den Kassettenrekorder in Stellung Aufnahme und drücken:
S7, 200, 237 (CR)
Der Computer antwortet nach der Aufnahme mit
READY
Die Startadresse von CHSBRD ist die vertraute Adresse 0200. Wir starten nun das Schachbrettprogramm:
200 (SP) R
Wenn uns kein Tippfehler unterlaufen ist, sieht das Ergebnis ungefähr so aus:

```
 X X X X
X X X X
 X X X X
X X X X
 X X X X
X X X X
 X X X X
X X X X
```

JUNIOR

3. ASCII-Tabelle — selbst gemacht!

Wie zu fast jedem Computer-Buch gehört auch zur Buchreihe über den Junior-Computer eine sogenannte ASCII-Tabelle: siehe Anhang 7. Dort sind sämtliche ASCII-Zeichen mit ihrem binären und hexadezimalen Kode aufgeführt. Eine solche auf Papier gedruckte Tabelle ist für das Entwickeln von Programmen unentbehrlich, die sich mit der Verarbeitung von ASCII-Zeichen beschäftigen. Man kann natürlich auch ein Programm aufstellen, das beim Drücken einer Taste auf der ASCII-Tastatur den zugehörigen binären und hexadezimalen Kode auf den Bildschirm setzt.
Während die beiden vorangegangenen Programmbeispiele nur von druckenden PM-Subroutinen Gebrauch machten, wird jetzt noch eine weitere Subroutine benötigt: Die Subroutine RECCHA, die auf das Drücken einer Taste wartet und den von der Tastatur gesendeten ASCII-Kode empfängt.

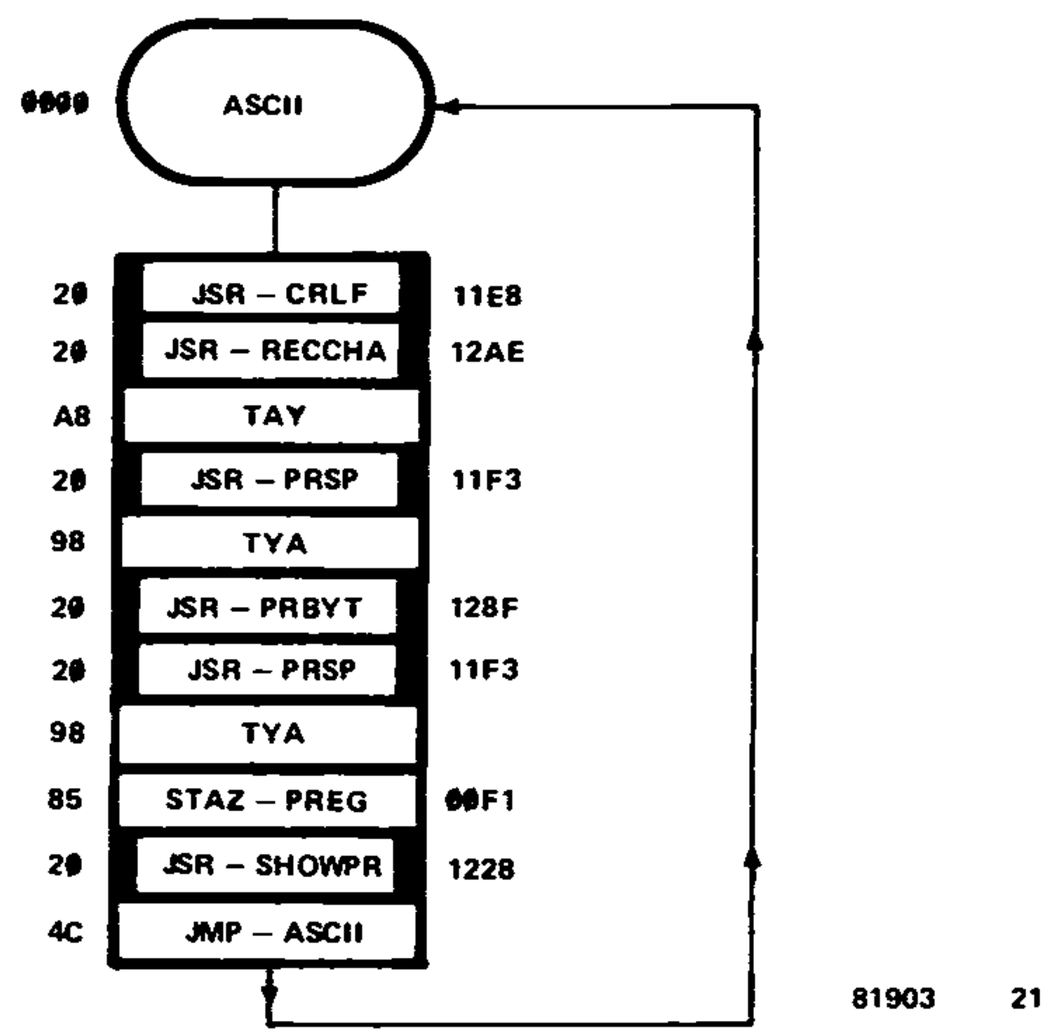


Bild 21. Das Programm ASCII. Hiermit kann durch Drücken einer Taste auf der ASCII-Tastatur der zugehörige ASCII-Kode sichtbar gemacht werden.

RECCHA ist die Abkürzung von RECeive CHAracter, was so viel wie "Zeichenempfänger" heißt. Der durch Drücken einer Taste erzeugte ASCII-Kode hängt übrigens auch noch davon ab, ob gleichzeitig die Taste SFT (Shift) oder die Taste CTRL (Control) betätigt wird. Nach Ablauf von RECCHA steht der ASCII-Kode der gedrückten Taste (lies: der gewählten Tastenfunktion) im Akku.
Das Programm für die Wiedergabe des ASCII-Kodes einer gewählten Tastenfunktion – es heißt natürlich ASCII! – ist in Bild 21 angegeben. Dieses aus 26 Byte bestehende Programm hat die Startadresse 0000. Am Anfang steht die Subroutine CRLF: beginne am Anfang einer neuen Zeile. Dann wartet das Programm auf das Drücken einer Taste, deren zugehörigen ASCII-Kode auf dem Bildschirm erscheinen soll. Sofern man eine Taste drückt, die zu einem sichtbaren Zeichen gehört, wird an den Zeilenanfang dieses Zeichen gesetzt.
Den im Akku stehenden ASCII-Kode der gewählten Tastenfunktion kopieren wir in das Y-Register (TAY). Das ist deshalb notwendig, weil beim folgenden JSR-PRSP der Inhalt des Akkus verlorengeht: Zwischen dem ASCII-Zeichen und seinem hexadezimalen Kode schieben wir zur besseren Übersichtlichkeit einen Zwischenraum ein. Die Wiedergabe des hexadezimalen ASCII-Kodes geschieht durch Rücktransport des ursprünglichen Akku-Inhalts aus dem Y-Register (TYA) und Aufruf der Subroutine PRBYT. Diese Subroutine wurde bereits von PRNUMB, dem ersten Programmbeispiel verwendet. PRBYT druckt zuerst das linke und dann das rechte Nibble des Akku-Inhalts aus, wobei die Werte der Nibbles zwischen 0 und F liegen können. Bevor die Nibbles zum Bildschirm wandern, wird jedes Nibble noch in den zugehörigen ASCII-Kode umgewandelt.
Dem ASCII-Kode in der hexadezimalen Form folgt auf dem Bildschirm

zunächst ein Zwischenraum. Um den ASCII-Kode anschließend bitweise darstellen zu können, setzen wir noch einmal den ursprünglichen Inhalt in den Akku (TYA). Dann kopieren wir den Akku-Inhalt in den inzwischen wohlbekannten Speicherplatz PREG (Adresse ØØF1) und springen zur Subroutine SHOWPR.
Weshalb wurde der Inhalt des Akkus nach PREG kopiert? Die Antwort ist einfach: Läßt man im Step-Modus von PM ein Listing von den Inhalten der Prozessor-Register ausdrucken (PM-Tastenfunktion L), dann wird der Inhalt des P-Registers bitweise wiedergegeben. Die bitweise Wiedergabe des dann in Speicherplatz PREG stehenden P-Register-Inhalts übernimmt die Subroutine SHOWPR. Bei Verwendung von PM-Subroutinen muß PREG aber zwangsläufig unbenutzt bleiben, da PM weder ganz noch teilweise Schritt für Schritt durchlaufen werden kann. Deshalb können wir in PREG einen beliebigen ASCII-Kode setzen, der dann als Bitmuster auf dem Bildschirm sichtbar gemacht wird.
Wenn sämtliche Bit des ASCII-Kodes ausgedruckt sind, folgt die Rückkehr von SHOWPR und der Sprung zum Programmanfang (JMP-ASCII). Nach Beginn einer neuen Zeile wartet das Programm auf die nächste gedrückte Taste (RECCHA). Man kann die Warteschleife durch Drücken von Taste RST und anschließendem neuen Start von PM oder durch Drücken der BRK-Taste während des Ausdruckens eines Zeichens verlassen.
Da das Programm ASCII nur Subroutinen von PM verwendet, das Hauptprogramm PM aber nicht läuft, erscheint auch keine Fehlermeldung "WHAT?" beim Drücken einer nicht zu PM gehörenden Taste (siehe Bild 12). Alle Tasten, die einen ASCII-Kode erzeugen, sind beim Programm ASCII in Funktion.
Die Eingabe des Programms ASCII aus Bild 21 ist schnell erledigt:

```
ASCII ØØØØ  2Ø  E8  11  JSR-CRLF
      ØØØ3  2Ø  AE  12  JSR-RECCHA
      ØØØ6  A8          TAY
      ØØØ7  2Ø  F3  11  JSR-PRSP
      ØØØA  98          TYA
      ØØØB  2Ø  8F  12  JSR-PRBYT
      ØØØE  2Ø  F3  11  JSR-PRSP
      ØØ11  98          TYA
      ØØ12  85  F1      STAZ-PREG
      ØØ14  2Ø  28  12  JSR-SHOWPR
      ØØ17  4C  ØØ  ØØ  JMP-ASCII
```

Das Programm ist zwar kurz, wir wollen es aber trotzdem auf Band setzen:
S8,,1A (CR)
READY
Wir starten das Programm:
(SP) R
Nun können wir nach Lust und Laune die Tastatur betätigen: Auf dem Bildschirm erscheint das betreffende Zeichen, gefolgt vom hexadezimalen und binären ASCII-Kode. In Tabelle 3 ist wiedergegeben, was ein Drucker dabei zu Papier brachte. Dort besteht der binäre Kode aus acht Bit. Das ganz links stehende Bit ist jedoch überall Ø; der eigentliche ASCII-Kode umfaßt nur sieben Bit. Beim Drücken der Taste SP (Hex-Kode 2Ø) wird

Tabelle 3. Hard Copy einer ASCII-Tabelle, ausgedruckt mit dem Programm ASCII aus Bild 21.

0	30	00110000	P	50	01010000	#	23	00100011
1	31	00110001	Q	51	01010001	$	24	00100100
2	32	00110010	R	52	01010010	%	25	00100101
3	33	00110011	S	53	01010011	^	5E	01011110
4	34	00110100	T	54	01010100	&	26	00100110
5	35	00110101	U	55	01010101	*	2A	00101010
6	36	00110110	V	56	01010110	(	28	00101000
7	37	00110111	W	57	01010111	)	29	00101001
8	38	00111000	X	58	01011000	_	5F	01011111
9	39	00111001	Y	59	01011001	_	5F	01011111
A	41	01000001	Z	5A	01011010	+	2B	00101011
B	42	01000010	,	2C	00101100	~	7E	01111110
C	43	01000011	.	2E	00101110	]	5D	01011101
D	44	01000100		20	00100000	|	7C	01111100
E	45	01000101	/	2F	00101111	:	3A	00111010
F	46	01000110	{	7B	01111011	"	22	00100010
G	47	01000111	'	27	00100111	}	7D	01111101
H	48	01001000	;	3B	00111011	<	3C	00111100
I	49	01001001	\	5C	01011100	>	3E	00111110
J	4A	01001010	[	5B	01011011	?	3F	00111111
K	4B	01001011	`	60	01100000		0D	00001101
L	4C	01001100	=	3D	00111101			
M	4D	01001101	-	2D	00101101		0A	00001010
N	4E	01001110	!	21	00100001			
O	4F	01001111	@	40	01000000			

kein Zeichen gedruckt. Auch die grafischen Kommandos CR (Hex-Kode ØD) und LF (Hex-Kode ØA) treten nicht durch irgendwelchen Zeichen, sondern nur durch ihre Wirkung in Erscheinung.
Das Programm ASCII wird sich beim folgenden vierten Programmbeispiel als nützlich erweisen, mit dem beliebige Texte auf den Bildschirm oder auf Papier geschrieben werden können.

4. Drucken von Texten

Das vierte und letzte Programmbeispiel in diesem Buch trägt den Namen FINAL; es ist in Bild 22 dargestellt. Warum dieses Programm FINAL heißt, wird vermutlich bald deutlich werden. Auch FINAL verwendet die PM-Subroutine RECCHA zum Empfang von ASCII-Zeichen sowie außerdem zwei weitere PM-Subroutinen.
Dank PM eignet sich der Junior-Computer hervorragend zum Ausdrucken von Texten, sei es auf dem Bildschirm, sei es mit Hilfe eines Druckers auf

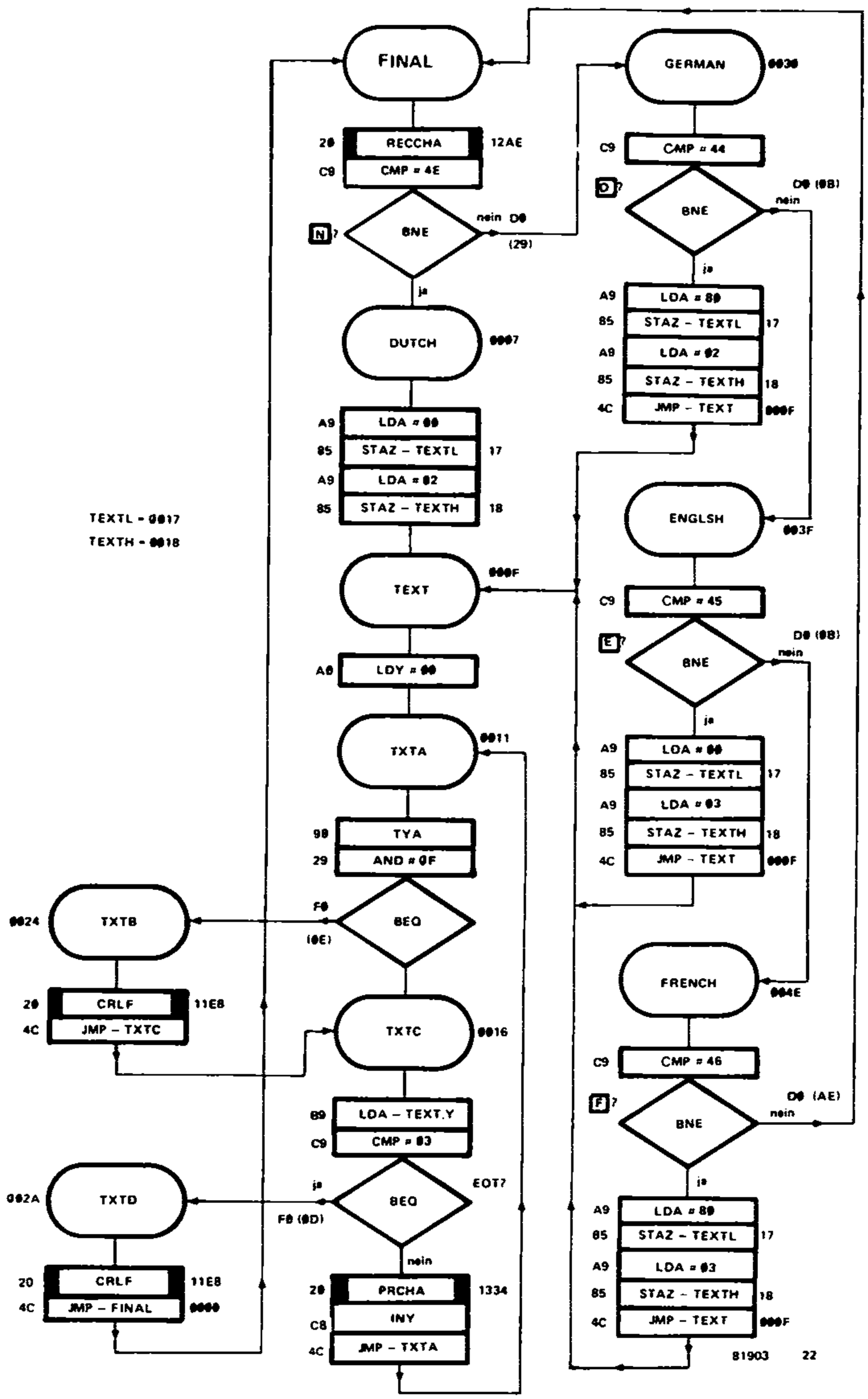


Bild 22. Das Programm FINAL schreibt auf Tastendruck wahlweise einen von vier Texten auf den Bildschirm, die in verschiedenen Sprachen abgefaßt sind.

Papier. Es sind hier nicht etwa nur Worte wie "JUNIOR" oder "WHAT?", sondern beliebige, vom Programmierer einzugebende Texte gemeint. Diese Texte können nach einmaliger Eingabe in den Computer und Konservierung auf Band beliebig oft ausgedruckt werden.

Das Programm FINAL ist eine Art internationaler Textmonitor. Die Tasten D (Deutsch), E (Englisch), F (Französisch) und N (Niederländisch) erhalten durch FINAL die in Klammern angegebenen speziellen Funktionen. Drückt man eine dieser Tasten, so schreibt der Junior-Computer einen bestimmten Text in der betreffenden Sprache auf den Bildschirm bzw. auf Papier. Die Texte selbst müssen vorher in das RAM gesetzt werden.
Weshalb vier Sprachen? Weil die Junior-Computer-Buchreihe in diesen vier Sprachen erscheint! Gleichzeitig macht das Programm FINAL deutlich, wie man eine oder mehrere Tasten der ASCII-Tastatur durch Software mit Sonderfunktionen belegen kann. Die Texte werden von FINAL so ausgedruckt, daß in jeder Zeile maximal 16 Zeichen oder Zwischenräume stehen. Als Textspeicher dienen die Seiten Ø2 und Ø3 des RAM auf der Basiskarte. Jeder Sprache ist ein gleich großer Abschnitt des Textspeichers zugeteilt:

N: Ø2ØØ . . . Ø27F; TEXTL=ØØ, TEXTH=Ø2
D: Ø28Ø . . . Ø2FF; TEXTL=8Ø, TEXTH=Ø2
E: Ø3ØØ . . . Ø37F; TEXTL=ØØ, TEXTH=Ø3
F: Ø38Ø . . . Ø3FF; TEXTL=8Ø, TEXTH=Ø3

Die Texte können folglich aus maximal 8 Zeilen mit jeweils 16 Zeichen bestehen. Da jeder Text mit einem sogenannten EOT-Zeichen (EOT = End of Text; ASCII-Kode Ø3) abgeschlossen sein muß, lassen sich je Sprache maximal 127 Zeichen speichern. Letzteres muß im ASCII-Kode geschehen; siehe hierzu Anhang 7.
Zuerst einige Erläuterungen zum eigentlichen "Druckwerk" von FINAL. Es beginnt beim Label TEXT mit dem Nullsetzen des Y-Registers. Der Y-Inhalt wird dazu in den Akku gesetzt (TYA) und das linke Nibble auf Null gesetzt (AND #ØF). Die Z-Flag stellt die Weiche für den weiteren Programmablauf. Sie entscheidet, ob vor dem Ausdrucken des nächsten Zeichens eine neue Zeile begonnen werden muß (CRLF).
Dann lädt das Programm den ASCII-Kode des nächsten auszudruckenden Zeichens in den Akku, wobei das Y-Register Hilfestellung leistet (absolute, Y-indizierte Adressierung des Textspeicherplatzes). Der Textbeginn wird vorher anhängig von der gewünschten Sprache durch Tastendruck (Tasten D. E, F und N) in den Speicherplätzen ØØ17 (TEXTL) und ØØ18 (TEXTH) spezifiziert. Wie schon erwähnt, müssen die Texte mit dem EOT-Zeichen Ø3 enden. Wenn dieses Zeichen auftritt, folgt ein Sprung zur PM-Subroutine CRLF (neuer Zeilenbeginn) und danach ein Sprung zum Anfang von FINAL. Bis dahin werden von PRCHA fortlaufend die im Textspeicher stehenden Zeichen ausgedruckt.
Die Tastenroutine RECCHA steht am Anfang von FINAL; sie setzt einen von der Tastatur empfangenen ASCII-Kode in den Akku. Mit mehreren CMPs und BNEs wird geprüft, ob der empfangene Kode zu einem der Zeichen D, E, F oder N gehört. Ist das nicht der Fall, so folgt ein Rücksprung nach FINAL; fällt der Test positiv aus, dann wird der Operand der Instruktion LDA-TEXT,Y entsprechend der gewählten Sprache in die Speicherplätze ØØ17 und ØØ18 gesetzt.
Das soll als Erläuterung von FINAL genügen. Wir geben dieses Programm jetzt in den Computer ein:

```
FINAL   ØØØØ  2Ø  AE   12  JSR-RECCHA
```

```
        0003  C9  4E        CMP #4E
        0005  D0  29        BNE GERMAN
DUTCH   0007  A9  00        LDA #00
        0009  85  17        STAZ-TEXTL
        000B  A9  02        LDA #02
        000D  85  18        STAZ-TEXTH
TEXT    000F  A0  00        LDY #00
TXTA    0011  98            TYA
        0012  29  0F        AND #0F
        0014  F0  0E        BEQ TXTB
TXTC    0016  B9  XX  XX    LDA-TEXT,Y
        0019  C9  03        CMP #03
        001B  F0  0D        BEQ TXTD
        001D  20  34  13    JSR-PRCHA
        0020  C8            INY
        0021  4C  11  00    JMP-TXTA
TXTB    0024  20  E8  11    JSR-CRLF
        0027  4C  16  00    JMP-TXTC
TXTD    002A  20  E8  11    JSR-CRLF
        002D  4C  00  00    JMP-FINAL
GERMAN  0030  C9  44        CMP #44
        0032  D0  0B        BNE ENGLISH
        0034  A9  80        LDA #80
        0036  85  17        STAZ-TEXTL
        0038  A9  02        LDA #02
        003A  85  18        STAZ-TEXTH
        003C  4C  0F  00    JMP-TEXT
ENGLSH  003F  C9  45        CMP #45
        0041  D0  0B        BNE FRENCH
        0043  A9  00        LDA #00
        0045  85  17        STAZ-TEXTL
        0047  A9  03        LDA #03
        0049  85  18        STAZ-TEXTH
        004B  4C  0F  00    JMP-TEXT
FRENCH  004E  C9  46        CMP #46
        0050  D0  AE        BNE FINAL
        0052  A9  80        LDA #80
        0054  85  17        STAZ-TEXTL
        0056  A9  03        LDA #03
        0058  85  18        STAZ-TEXTH
        005A  4C  0F  00    JMP-TEXT
```

Ein IRQ-Sprungvektor braucht nicht spezifiziert zu werden, da im Programm FINAL kein BRK-Befehl vorkommt. Der Programmablauf mündet stets in die Warteschleife für den Empfang eines von der Tastatur gesendeten ASCII-Kodes ein. FINAL kann durch Drücken der Taste RST verlassen werden; über 1 0 0 0 GO RUBOUT kann man zu PM zurückkehren. Außerdem kann man FINAL durch Betätigen der BRK-Taste während des Textausdruckens abbrechen. Eine weitere Möglichkeit ist das Drücken der Taste ST der Standard-Tastatur. Der BRK- bzw. NMI-Sprungvektor muß dabei auf ein geeignetes Ziel gerichtet sein. Diese Vektoren weisen automa-

tisch auf die Startadresse von PM, wenn PM nach dem Einschalten des Junior-Computers irgendwann gestartet wurde. Das aber ist wahrscheinlich, denn mit PM gestaltet sich die Eingabe oder das Einlesen von Programmen bequemer als mit dem Standard-Monitor bzw. mit TM.
Hier nun die Textdaten. Wir beginnen mit dem deutschen Text:

```
M
HEXDUMP: 280,2BF
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0280:  4C 49 45 42 45 52 20 4C 45 53 45 52 21 44 41 53
0290:  57 41 52 20 45 53 20 46 55 45 52 20 48 45 55 2D
02A0:  54 45 2E 41 55 46 20 57 49 45 44 45 52 53 45 2D
02B0:  48 45 4E 20 49 4E 20 42 55 43 48 20 34 21 21 03
02C0:
JUNIOR
```

Man vergesse vor allem das EOF-Zeichen 03 in Speicherplatz 02BF nicht! Fehlt dieses Zeichen, dann werden nach dem "echten" Text willkürliche Daten ausgedruckt. Dabei besteht die Gefahr, daß dies unendlich lange andauert (wenn nicht zufällig irgendwo 03 steht). Ein Druck auf die BRK-Taste macht dem Spuk ein Ende.
Es folgt der englische Text:

```
M
HEXDUMP: 300,340
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0300:  44 45 41 52 20 4D 52 2F 4D 52 53 2F 4D 49 53 53
0310:  4A 55 4E 49 4F 52 21 49 54 27 53 20 54 49 4D 45
0320:  54 4F 20 47 4F 2E 53 45 45 20 59 4F 55 20 20 20
0330:  41 47 41 49 4E 20 49 4E 20 42 4F 4F 4B 20 34 21
0340:  03
JUNIOR
```

Der französische Text ist etwas länger:

```
M
HEXDUMP: 380,3E0
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0380:  43 48 45 52 53 20 4C 45 43 54 45 55 52 53 2C 20
0390:  43 27 45 53 54 20 46 49 4E 49 20 50 4F 55 52 20
03A0:  41 55 4A 4F 55 52 44 27 48 55 49 21 20 20 20 20
03B0:  52 45 4E 44 45 5A 2D 56 4F 55 53 20 41 55 58 20
03C0:  50 52 45 4D 49 45 52 45 53 20 50 41 47 45 53 20
03D0:  44 55 20 4C 49 56 52 45 20 51 55 41 54 52 45 20
03E0:  03
JUNIOR
```

Und zum Schluß der niederländische Text:

```
M
HEXDUMP: 200,230
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0200: 42 45 53 54 45 20 4D 45 4E 53 45 4E 21 44 49 54
0210: 57 41 53 20 48 45 54 20 44 41 4E 2E 54 4F 54 20
0220: 5A 49 45 4E 53 20 49 4E 20 42 4F 45 4B 20 34 21
0230: 03
```

JUNIOR

Nach so viel Tipparbeit setzen wir zuerst einmal alles auf Band. Wir starten den Kassettenrekorder in Stellung Aufnahme und geben in den Computer ein:

S2,,5D (CR)
READY (Pausentaste drücken)
S3,200,231 (CR)
READY (Pausentaste drücken)
S4,380,3E1 (CR)
READY (Pausentaste drücken)
S5,280,2C0 (CR)
READY (Pausentaste drücken)
S6,300,341 (CR)
READY (Kassettenrekorder stoppen)

Nun endlich wird FINAL gestartet und zuerst der deutsche Text sichtbar gemacht:

(SP) R D

Um einen anderen Text ausdrucken zu lassen genügt es, eine der Tasten E, F und N zu drücken.

Den Schlußsatz zu diesem Buch hat damit der Computer selbst geschrieben.

LIEBER LESER!DAS WAR ES FUER HEU-TE.AUF WIEDERSE-HEN IN BUCH 4!!

Basiskarte und RAM/EPROM-Karte

Speichererweiterung ohne Interfacekarte

Es sind Anwendungen des Junior-Computers denkbar, bei denen nur eine Speichererweiterung notwendig ist, jedoch auf die übrigen Funktionen der Interfacekarte verzichtet werden kann. An Stelle der Interfacekarte läßt sich dann vorteilhaft eine einzelne RAM/EPROM-Karte verwenden; sie kann mit der Basiskarte direkt verbunden werden. Der unmittelbare Anschluß von mehr als einer RAM/EPROM-Karte ist nicht möglich, da der Adreß- und Datenbus auf der Basiskarte nicht gepuffert ist.
Hier die Bauanleitung:

① **Die Drahtbrücke auf der Basiskarte wird von D nach EX gelegt.** Nun müssen nämlich nicht nur Bausteine auf der Basiskarte, sondern auch auf der anzuschließenden RAM/EPROM-Karte adressiert werden. Die freien Selektionssignale K1 . . . K5 lassen sich dafür nicht verwenden.

② **Auf der RAM/EPROM-Karte wird Pin 3 von IC29 oder Pin 4 von IC8** (Ausgang von N5; siehe Bild 25 in Kapitel 10) **mit dem Punkt EX verbunden** (Pin 30c des Erweiterungskonnektors und des Anschlußkonnektors der RAM/EPROM-Karte).

Das hat folgenden Grund: Das Ausgangssignal von N5 wird durch die Eingangssignale von N1 und N2 bestimmt. Diese liegen an den Anschlüssen X, Y, V und W, von denen einer oder mehrere mit den Ausgängen von Hauptadressendekoder IC5 verbunden werden. Solange kein Eingang von N1 und N2 logisch Ø ist, sind die Ausgänge von N1 und N2 logisch 1; der Ausgang von N5 ist folglich logisch Ø. EX ist daher logisch Ø, so daß der Adressendekoder IC6 auf der Basiskarte wie bisher arbeitet: es können Bausteine auf der Basiskarte selektiert und adressiert werden. Wenn einer der Eingänge von N1 oder N2 auf logisch Ø geht, weil ein 4-K-Adressenblock auf der RAM/EPROM-Karte selektiert wird, ist der Ausgang von N5 und damit auch der D-Eingang von IC6 auf der Basiskarte logisch 1. Nun können Bausteine adressiert werden, die nicht auf der Basiskarte sitzen (und auch nicht auf der Interfacekarte; sie spielt hier keine Rolle). Zur Arbeitsweise des Adressendekoders auf der Basiskarte siehe Bild 4 und Tabelle 1 von Kapitel 10.
Daraus folgt:

③ **Die Anschlußpunkte "Ø", "1" und – wenn die Vektoren aus dem EPROM auf der Basiskarte geholt werden müssen – "F" des Hauptadressendekoders dürfen nicht beschaltet werden!** Würde man dies tun, dann ginge das Signal EX bei den Adressen ØXXX, 1XXX und eventuell

FXXX auf logisch 1, so daß die für diese Adressen notwendige Selektion auf der Basiskarte (mit IC6) nicht zustande käme.
Abgesehen davon gibt es einen zweiten Grund, weshalb die Anschlußpunkte "Ø" und "1" frei bleiben müssen. Angenommen man will das Loch des Adressenbereichs zwischen ØØØØ und 17FF (Bausteine auf der Basiskarte) mit Speichern auf der RAM/EPROM-Karte füllen (der Bereich 18ØØ . . . 19FF läßt sich nicht füllen, da dieser zum Selektionssignal K6 gehört). Es gibt dann stets einen 4-K-Bereich, der teilweise zu Speichern auf der RAM/EPROM-Karte und teilweise zu Speichern auf der Basiskarte gehört. Wird aber ein Speicher auf der Basiskarte gelesen, der innerhalb eines derart geteilten 4-K-Bereichs liegt, dann werden dadurch die Datenbuspuffer der RAM/EPROM-Karte in Leserichtung aktiv: Es tritt die unter allen Umständen zu verhindernde Situation ein, bei der Daten verfälscht werden können (siehe Kapitel 10, Bild 6).
④ In dem Falle, daß die Vektoren NMI, RES und IRQ aus dem EPROM auf der Basiskarte geholt werden müssen, braucht und darf Anschluß "F" von IC 5 auf der RAM/EPROM-Karte nicht beschaltet werden. Die Adreßleitungen A13, A14 und A15 sind dann ohne Funktion, so daß der Prozessor die Vektoren aus dem Basiskarten-EPROM liest (Seite 1F an Stelle von FF). **Der Bereich FØØØ . . . FFFF kann nicht mit Speicherbausteinen der RAM/EPROM-Karte belegt werden.** Für eine einzelne RAM/EPROM-Karte bleibt jedoch ein genügend weiter Bereich übrig: die Seiten 2Ø . . . EF. Dieser 52-K-Bereich ist auf jeden Fall bedeutend größer als der 24-K-Bereich, den eine RAM/EPROM-Karte maximal ausfüllen kann (8 K RAM und 16 K EPROM).
⑤ Wenn die Vektoren aus einem EPROM auf der RAM/EPROM-Karte geholt werden, **muß Anschluß "F" beschaltet werden. Natürlich muß Seite FF auch mit einem EPROM belegt sein, das die Vektoren enthält.**
Das betrifft die Adressen
FFFA für NMIL
FFFB für NMIH
FFFC für RESL
FFFD für RESH
FFFE für IRQL
FFFF für IRQH
Der mit Speicherbausteinen auf der RAM/EPROM-Karte belegbare Adressenbereich umfaßt die Seiten 2Ø . . . FF. Näheres ist dem letzten Teil von Kapitel 10 zu entnehmen, in dem die RAM/EPROM-Karte besprochen wird.
Noch eine Frage ist zu klären: Wie sieht es mit der **achtfachen Adressierung** aus? Beim Junior-Computer in seiner Basisversion fallen bekanntlich die Seiten ØX mit den Seiten 2X, 4X, 6X, 8X, AX, CX und EX sowie die Seiten 1X mit den Seiten 3X, 5X, 7X, 9X, BX, DX und FX zusammen (X = Ø . . . F).
Nun, die bisherige Adressendekodierung wird beibehalten, solange die Adresse **nicht** im Bereich eines 4-K-Adreßblocks liegt, der auf der RAM/EPROM-Karte dekodiert wird. Ausschlaggebend ist die Anzahl der 4-K-Blöcke (also der IC5-Anschlüsse), die auf der RAM/EPROM-Karte belegt sind, wenn es um die Frage der Mehrfachadressierung geht. Achtfach ist sie in jedem Fall nicht mehr. Wenn die Vektoren aus dem EPROM der

Basiskarte gelesen werden, ist bei den Seiten 1X (X = Ø . . . F) von einer mindestens zweifachen Adressierung die Rede.

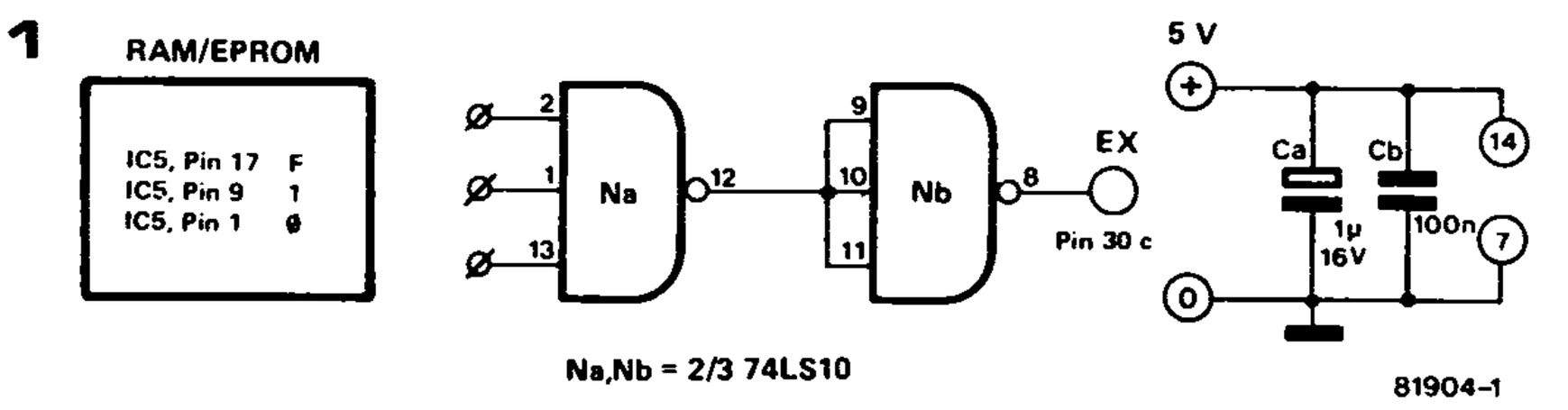
⑥ Die **Verbindungen** zwischen den beiden Karten können aus kurzen Leitungen, aus Konnektoren und (einem Teil) der Buskarte oder aus Konnektoren kombiniert mit Flachkabel bestehen. Die Bilder 19 und 20 in Kapitel 10 geben Hilfestellung für den Aufbau.

⑦ Für die **Speisung** einer einzelnen RAM/EPROM-Karte neben der Basiskarte reicht das Netzteil in seiner Grundversion aus. Eventuell muß die Kühlung der Stabilisator-ICs verbessert werden. Das 2708 benötigt neben +5 V auch noch die Spannungen –5 V und +12 V, das 2716 begnügt sich ebenso wie die RAMs mit +5 V. Daher sind insbesondere bei der Bestückung mit 2716-EPROMs keine Probleme mit dem nicht erweiterten Netzteil zu erwarten.

Erste Alternative

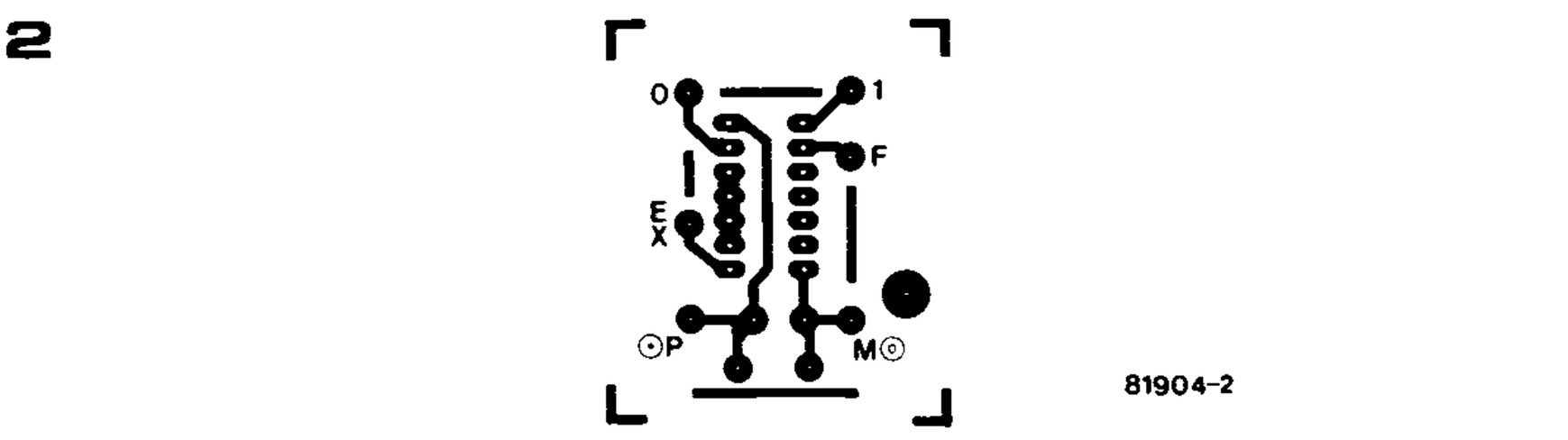
Als Vorbereitung für die Erweiterung des Junior-Computers mit einer einzelnen RAM/EPROM-Karte wurde der Ausgang von N5 auf der RAM/EPROM-Karte über EX mit Eingang D von IC6 auf der Basiskarte verbunden. Mit der Schaltung in Bild 1 geht es jedoch auch anders. Sie hat den

1



Vorteil, daß die achtfache Adressierung vollständig aufgehoben wird (mit Ausnahme der zweifachen Adressierung beim Lesen der Vektoren aus dem Basiskarten-EPROM; die Seiten 1X müssen hierbei mit den Seiten FX identisch sein). Ein Nachteil ist die etwas schwierigere Realisierung in der Praxis: Entweder baut man die Schaltung auf der Platine (Bild 2) oder auf einer Lötpunktrasterplatte auf.

2



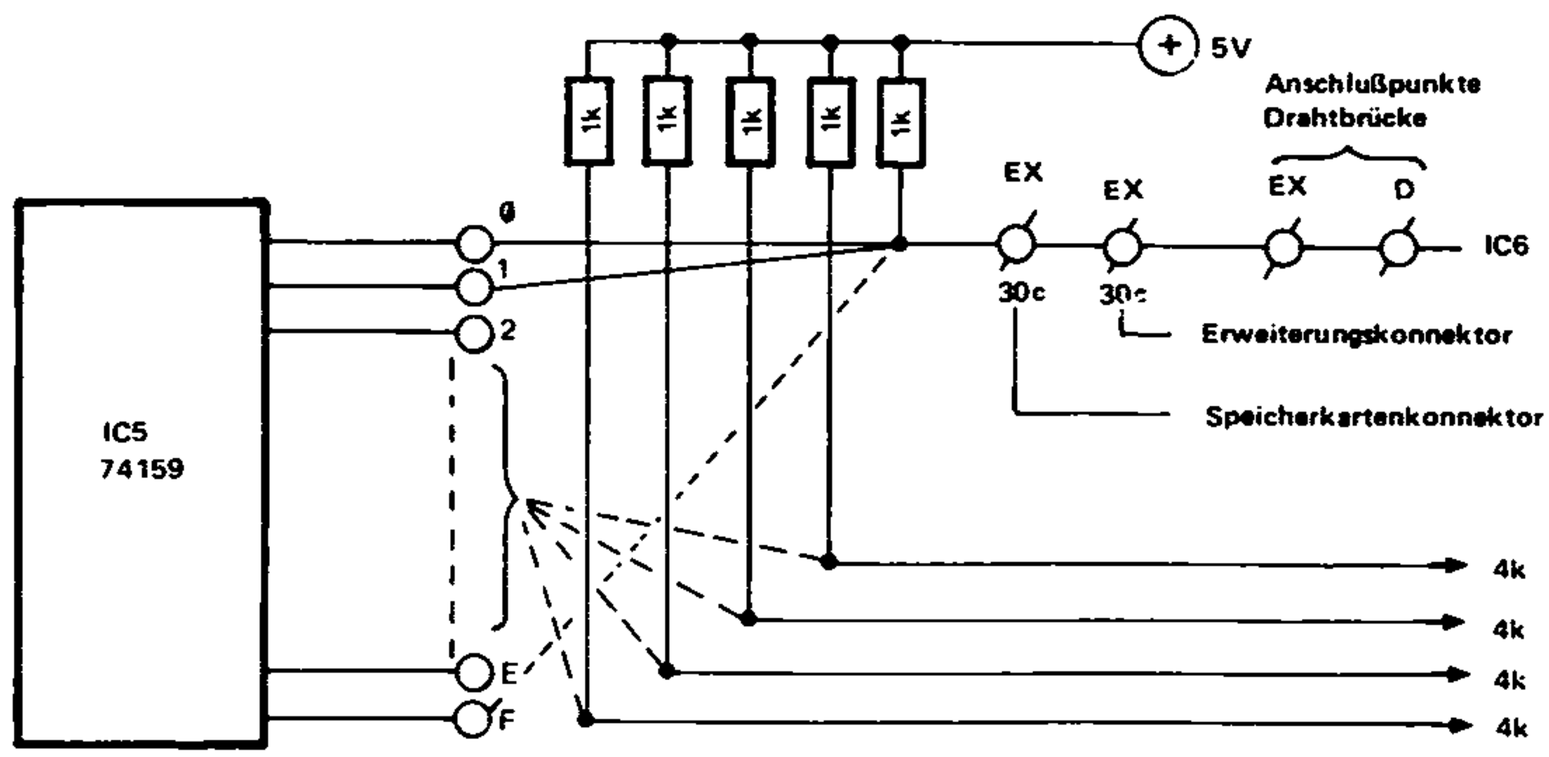
Die Verbindung mit Anschlußpunkt "F" von IC5 entfällt, wenn die Vektoren aus dem EPROM auf der Basiskarte geholt werden. Auch die Anschlüsse Ø und 1 sollten nicht mit X, Y, V oder W verbunden werden, da die zugehörigen 4-K-Adreßblöcke nicht vollständig belegt werden können

(ein Teil davon ist bereits auf der Basiskarte belegt) und die Eigenschaften der Datenbuspuffer Anlaß zu Störungen sein können. Im übrigen behalten die vorangegangenen Hinweise ihre Gültigkeit.

Zweite Alternative

Neben den beiden beschriebenen Hardware-Modifikationen gibt es noch eine dritte Möglichkeit; diese ist in Bild 3 dargestellt. Als Hauptadressendekoder (IC5) auf der RAM/EPROM-Karte wird ein 74159 an Stelle des 74154 verwendet. Dieser ist bis auf die offenen Kollektorausgänge mit dem ursprünglichen Typ identisch. Das bedeutet, daß die Ausgänge Ø . . . F

3

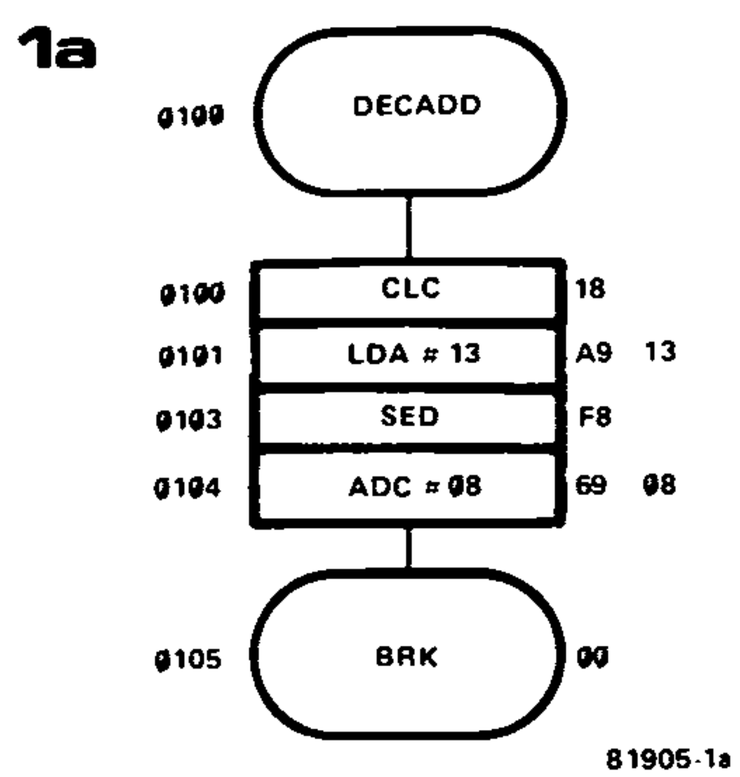


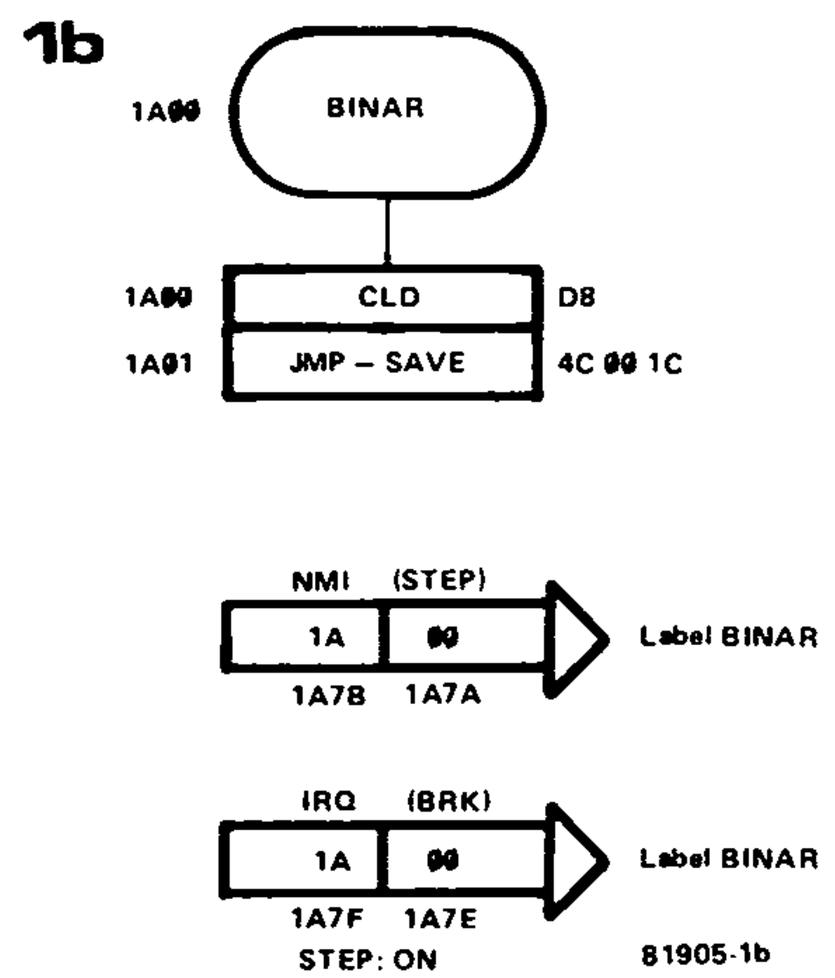
über Pull-Up-Widerstände an +5 V gelegt werden müssen, um definierte logische Signale zu erhalten. Hier dürfen jedoch mehrere Ausgänge direkt miteinander verbunden werden, was beim 74154 wegen seiner Gegentaktausgänge nicht möglich war.
Wenn man die Ausgänge Ø und 1 von IC5 miteinander verbindet und an Eingang D von IC6 auf der Basiskarte legt (über die EX-Leitung), gehören ausschließlich die Adressen ØØØØ . . . 1FFF zur Basiskarte. Übrigens: Nur ein Teil dieser Adressen ist tatsächlich belegt. Auf jeden Fall wird die achtfache Adressierung beseitigt.
Soll der Prozessor die Vektoren aus dem Basiskarten-EPROM lesen, dann wird Anschlußpunkt "F" in gleicher Weise verbunden. Ein Pull-Up-Widerstand ist an jedem Anschlußpunkt 2 . . . F (bzw. 2 . . . E) notwendig, der für die Dekodierung eines mit Speicherbausteinen belegten 4-K-Adreßblocks benutzt wird.

PIA-RAM-Erweiterung für den Standard-Monitor

Schritt für Schritt dezimal rechnen

Angenommen wir machen keinen Gebrauch von Printer Monitor (PM), sondern benutzen den Standard-Monitor. Wir wollen ein Programm in Einzelschritten ausführen lassen, in dem dezimal gerechnet wird und daher der Befehl SED (F8) vorkommt. Nehmen wir als Beispiel das Programm DECADD in Bild 1a.





Wir geben die Daten ein, richten den NMI- und den IRQ-Sprungvektor auf 1C00 (SAVE-Eingang des Monitors), setzen den STEP-Schalter auf ON und geben über das Keyboard folgendes ein:

Tasten	Display		
AD 0 1 0 0 GO	0101	A9	(CLC ausgeführt)
(Ausstieg) PC	0101	A9	
GO	0103	F8	(LDA ausgeführt)
(Ausstieg) PC	0103	F8	
GO	0104	69	(SED ausgeführt)
(AD) 0	1040	XX	
0	0400	XX	
F	0400	XX	????
1	4001	XX	

Wir wollten nach Ausführung des Befehls SED (Set Decimal Flag) den Inhalt des P-Registers (ØØF1) sichtbar machen, um zu prüfen, ob die D-Flag tatsächlich 1 geworden ist. Taste AD ist in Klammern gesetzt, **da der Adreßmodus bei Eintritt über SAVE oder RESET in den Monitor automatisch eingeschaltet ist.** Offensichtlich reagiert der Monitor nicht auf Drücken von Taste F und auch nicht auf Drücken von Taste PC, wenn man beschließt, ohne Kontrolle des P-Registers fortzufahren.

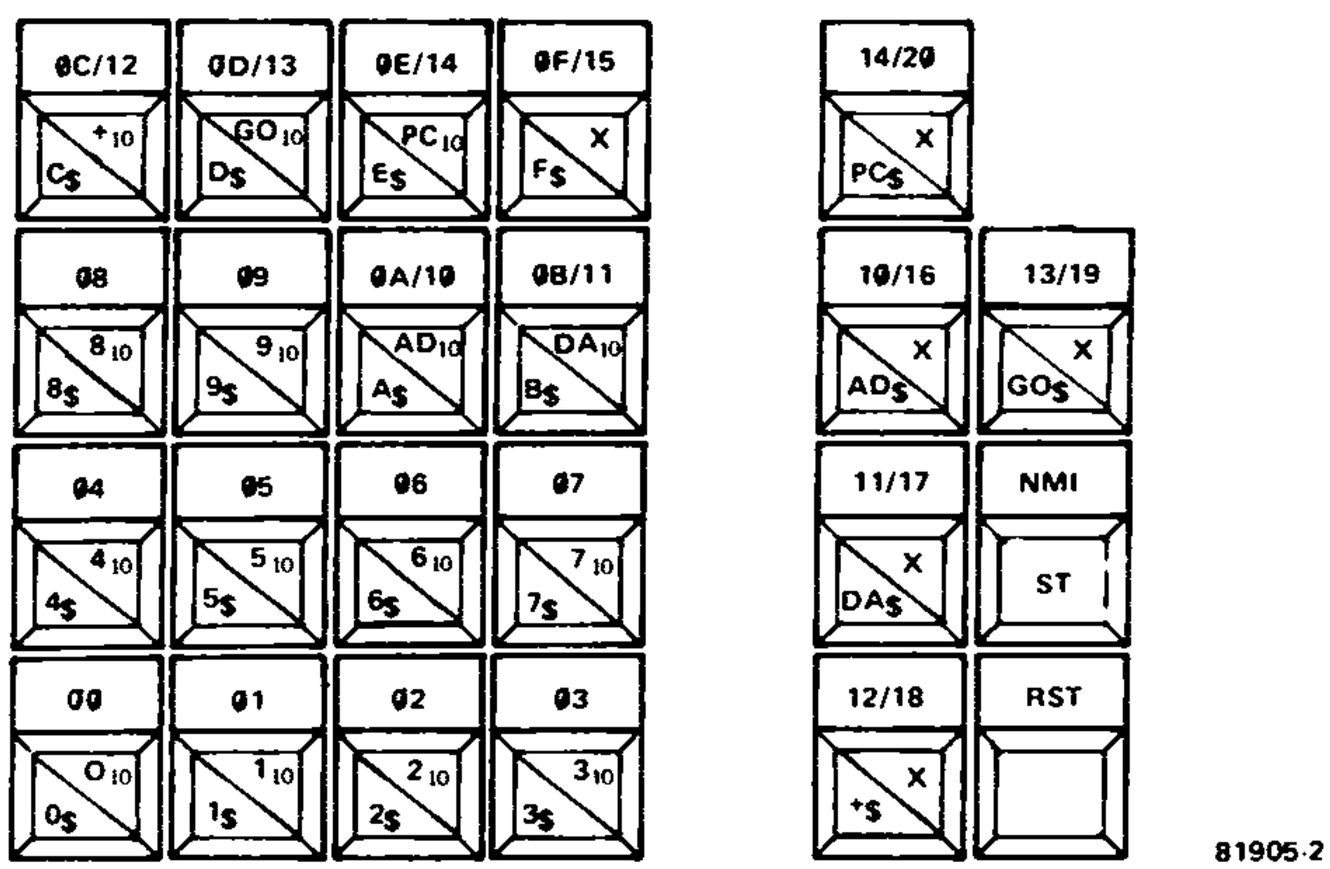
Was ist geschehen?

Nach Ausführen des Befehls SED rechnet der Prozessor dezimal; für die richtige Arbeitsweise des Monitors ist jedoch notwendig, daß der Prozessor hexadezimal rechnet. Deshalb steht auch am Anfang von RESET der Befehl CLD (siehe Bild 3a), **am Anfang der SAVE-Routine jedoch nicht!** Grundsätzlich spricht nichts dagegen, auch hier einen CLD einzufügen, denn der Inhalt des P-Registers wird ja bei jedem Sprung aus einem Anwenderprogramm zu SAVE gerettet. Nach dem Rücksprung vom Monitor zum Programm wird er während der Ausführung des Programmteils GOEXEC des Monitors wiederhergestellt (nach Drücken von GO; siehe Kapitel 7 in Buch 2). Ohne weiteres darf deshalb ein Flag während des Verbleibs im Monitor verändert werden; später wird alles automatisch in seinen ursprünglichen Zustand gebracht.

Folgen

Der Monitor ist nach SED nicht völlig funktionsunfähig. Wie Bild 2 zeigt, erhalten verschiedene Tasten eine andere Funktion einschließlich der Funktionslosigkeit (Tasten F, AD, DA, +, GO und PC). Bei den Tasten RST und ST ändert sich nichts, die numerischen Tastenfunktionen A . . . F sind dagegen verlorengegangen. Der Index Dollarzeichen kennzeichnet die normale Funktion der betreffenden Taste, während der Index 10 auf die Funktion nach SED hindeutet.

2



Erklärung

Der Monitor arbeitet auf der Basis des Wertes der gedrückten Taste, siehe Buch 2, Seite 142, Bild 16 mit dem zugehörigen Text. Die Tastenwerte werden nun nicht mehr durch hexadezimale, sondern durch dezimale Rechenoperationen ermittelt, was, wie Bild 2 zeigt, eine Änderung der Funktion für einen Teil der Tasten zur Folge hat. Die Steuerfunktionen liegen jetzt auf anderen Tasten, während die numerischen Tastenfunktionen A . . . F nicht mehr zugänglich sind. Letzteres bedeutet, daß während des Verbleibs im Monitor keine Adressen eingegeben werden können, die die Ziffern A . . . F enthalten. Man könnte höchstens die nächstniedrigere Adresse eingeben, die nur die Ziffern Ø . . . 9 enthält, und dann die Plustaste (jetzt Taste C!) so oft drücken, bis die gewünschte Adresse erreicht ist. Dies ist natürlich mehr als umständlich. Ferner wird die Bedienung durch das Verlagern der Steuerfunktionen auf die Tasten A . . . F nicht gerade erleichtert.

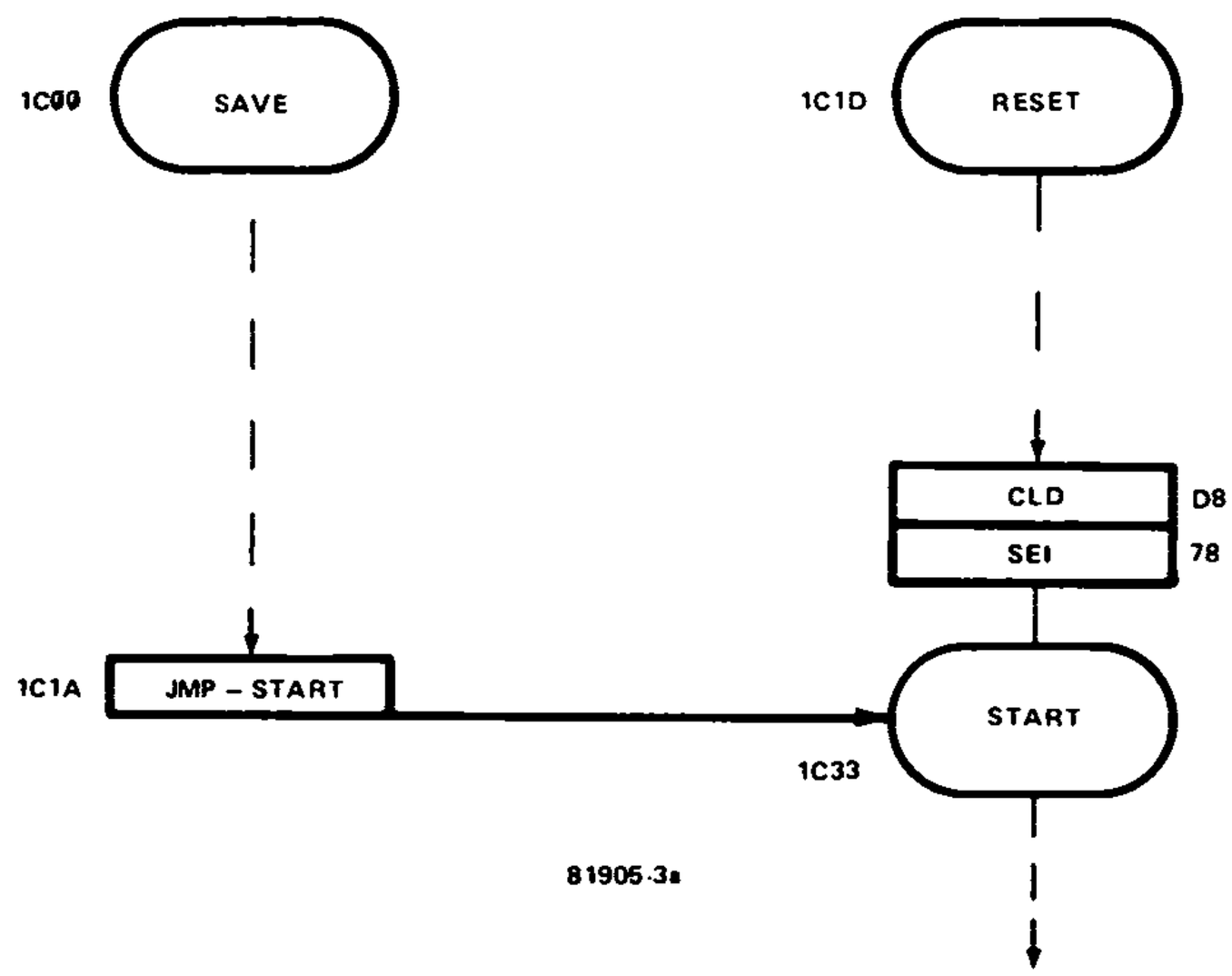
Erste Lösung: Änderung des EPROM-Inhalts

Alle diesbezüglichen Probleme sind beseitigt, wenn auch am SAVE-Eingang des Monitors ein CLD-Befehl steht, so daß Bild 3b an die Stelle von Bild 3a tritt. Das Ganze läuft auf das Verschieben des Label START und das Wechseln der Befehle SEI und CLD hinaus. Der letzte Befehl gehört dann nicht mehr zur Resetroutine, sondern zum Hauptprogramm des Monitors, das bei START beginnt. Der Inhalt folgender drei EPROM-Speicherplätze muß geändert werden:
IC18 → 32 (bisher 33)
IC31 → 78 (bisher D8)
IC32 → D8 (bisher 78)
Dies läßt sich natürlich nur durch Neuprogrammierung des EPROMs bewerkstelligen. Vom Elektor-Software-Service wird jedoch unter der Nummer ESS 503 weiterhin der Monitor in seiner unveränderten Fassung geliefert. Es gibt nämlich noch eine andere Lösung des Problems.

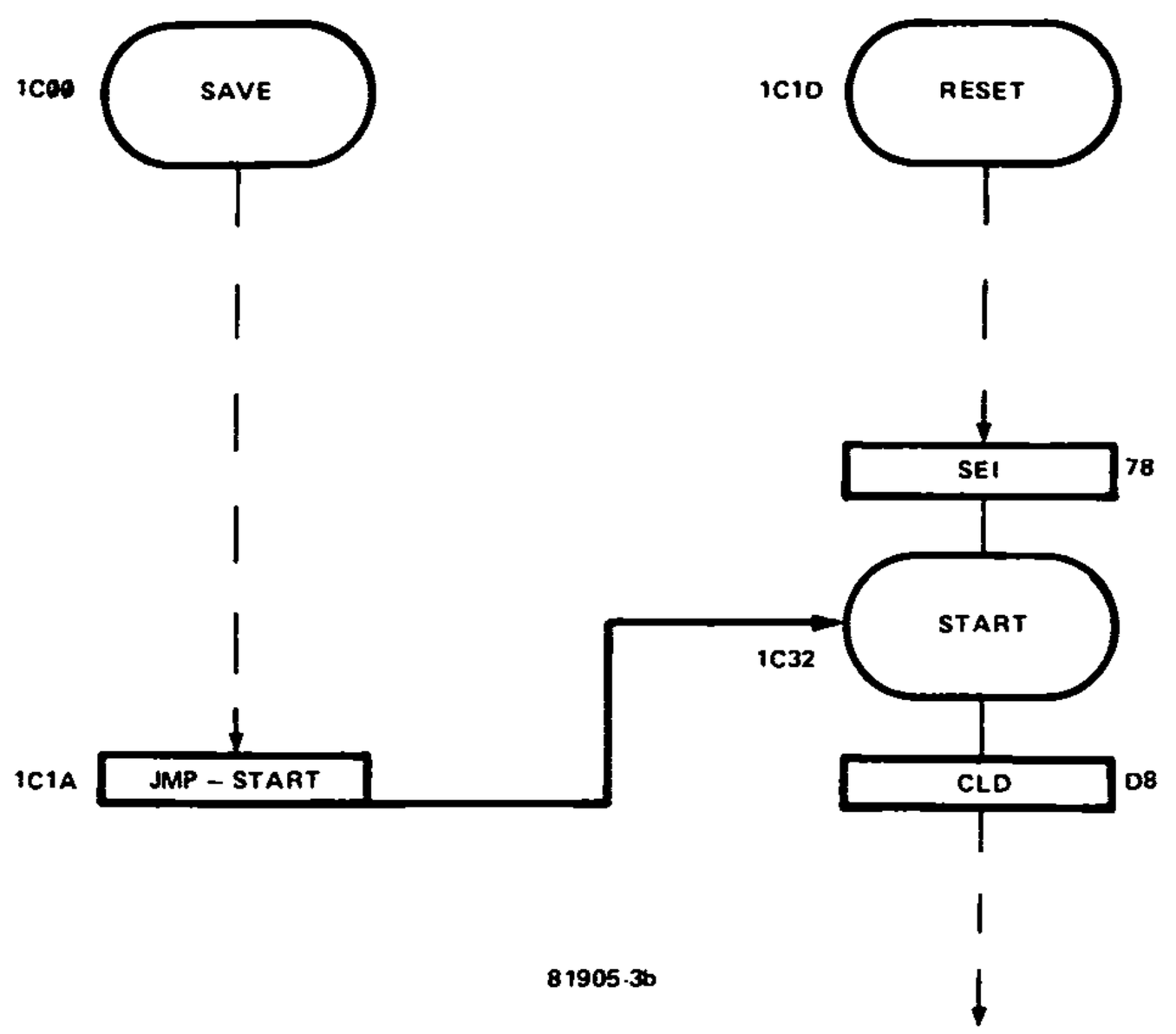
Zweite Lösung: Monitor-Ergänzung im PIA-RAM

Was ist zu tun?
① Die Aufsteckplatine aus Kapitel 10 (Bild 8b, 11 und 12 jenes Kapitels) wird auf die Basisplatine des Junior-Computers montiert. Einzelheiten können dem zu diesen Bildern gehörenden Text entnommen werden.
② Der Anschluß K7 von Adreßdekoder IC6 auf der Basiskarte wird mit einem K-Anschluß der Aufsteckplatine verbunden, der K6-Anschluß des Adreßdekoders (Widerstand R15 bietet einen Verbindungspunkt) wird an den anderen K-Anschluß gelegt.
③ Man gibt das Programm BINAR aus Bild 1b ein (BINAR = BINary ARithmetic), und zwar auf Seite 1A oder, was damit identisch ist, auf Seite 1B. Da es sich hier um einen RAM-Bereich handelt, müssen die beiden Instruktionen aus Bild 1b nach jedem Einschalten des Juniors neu eingegeben werden (natürlich nur, wenn der Computer dezimal rechnen soll!).

3a



3b



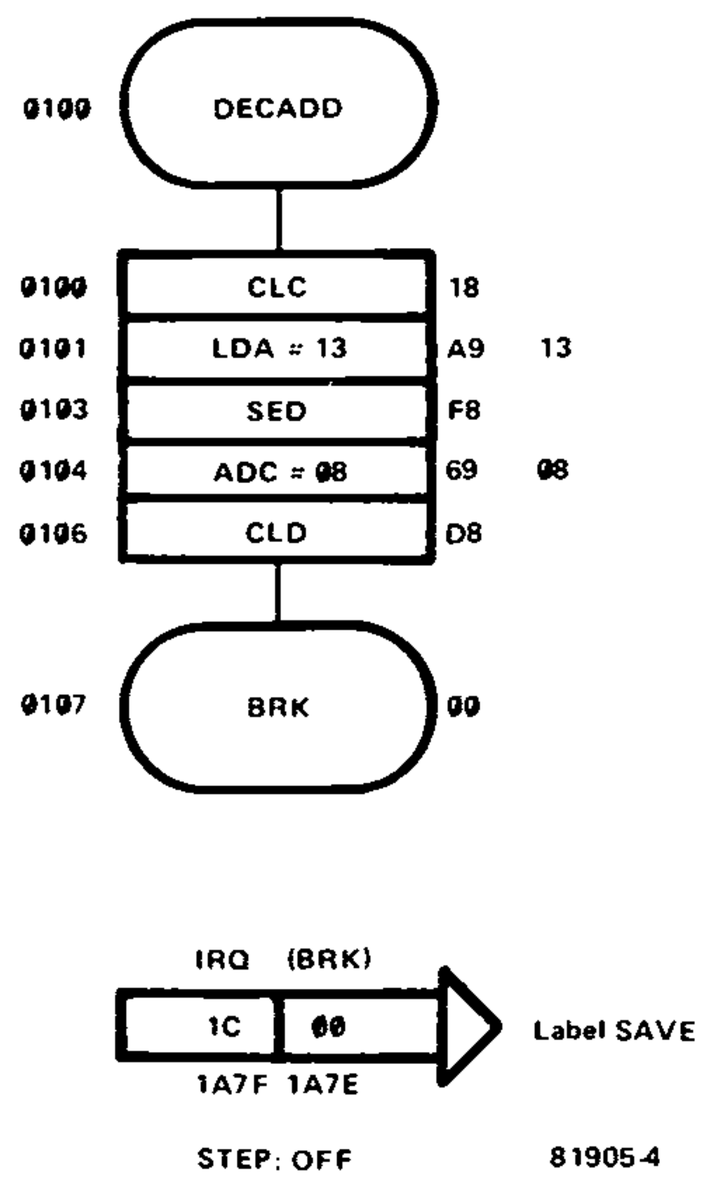
④ Der NMI-Vektor wird auf die Startadresse von BINAR gerichtet (Step-by-Step-Mode). Endet das Programm wie in Bild 1a mit BRK, dann muß auch der IRQ-Vektor auf BINAR zeigen.

⑤ Man setzt den STEP-Schalter auf ON, gibt die Startadresse ein und drückt GO. Die Beschränkungen beim schrittweisen Abarbeiten eines Programms wie Bild 1a sind damit beseitigt; alle Tasten haben ihre normale Funktion.

Übrigens:

Auch beim ununterbrochenen Abarbeiten des Programms in Bild 1a, bei dem der IRQ-Vektor (BRK) auf 1CØØ zeigt, ist der Monitor ohne die beschriebene Maßnahme in seiner Funktion eingeschränkt; nach BRK ist nämlich die D-Flag unverändert logisch 1. Für den **ununterbrochenen Durchlauf eines dezimalen Rechenprogramms** gibt es natürlich noch eine andere Lösung. Man sorgt für den Übergang von der dezimalen zur hexadezimalen Rechenweise nach der Addition und vor dem BRK, also vor dem Sprung zum Monitor. Dies ist in Bild 4 dargestellt. Jetzt kann allerdings das Programm ohne weitere Maßnahmen nicht in voller Länge schrittweise durchlaufen werden.

4



Was haben die Schritte ① . . . ⑤ bewirkt? Wir haben die Umstellung auf binäres Rechnen (CLD) an die SAVE-Routine angefügt und den schrittweisen Durchlauf eines im PIA-RAM stehenden Programms blockiert; siehe dazu auch den betreffenden Abschnitt von Kapitel 10 sowie Bild 9. Daß die Aufsteckplatine wirklich erforderlich ist, beweist das folgende Experiment: Streichen wir die Punkte ① und ② und führen nur die übrigen Punkte aus, dann führt beim schrittweisen Durchlaufen des Programms in Bild 1a zwar der erstfolgende NMI nach BINAR (1AØØ), die zugehörige Instruktion bleibt jedoch unerledigt, da nur eine einzige Instruktion gleichzeitig ausgeführt wird. Das aber war die Instruktion des Programms, die den NMI auslöste. Es folgt deshalb kein Sprung nach SAVE, also zum Monitor; der Ausstieg ist nicht möglich und die Funktion der GO-Taste für die Ausführung der nächsten Instruktion ist unzugänglich. Die Displays sind abgeschaltet.
Deshalb wird hier entweder die Aufsteckplatine oder die geänderte Monitor-Fassung benötigt.

Anhang 3

Vektoren und Buskartenspeicher

Alternativen ohne EPROM

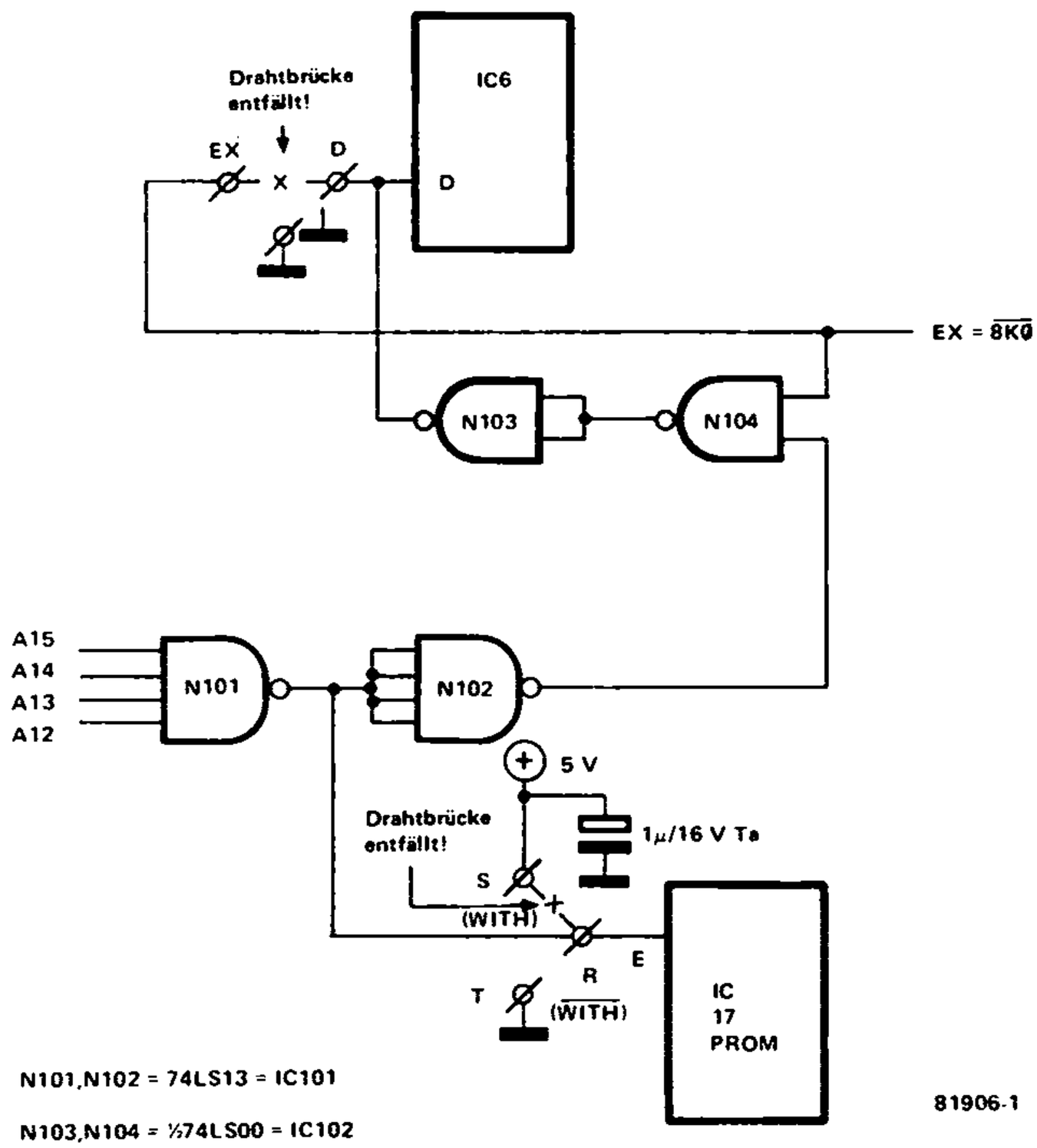
Wie in Kapitel 10 beschrieben, müssen auf Seite FF unter den Adressen FFFA . . . FFFF die Sprungvektoren stehen, sobald der Junior-Computer durch eine oder mehrere Busspeicherkarten erweitert wird. Dies zu realisieren kann unverhältnismäßig aufwendig sein. Wenn der Junior nur zusätzliche RAM-Speicherkapazität erhalten soll, zum Beispiel durch dynamische 16-K-RAM-Karten (nicht in diesem Buch beschrieben), dann ist allein wegen der sechs Vektoren-Bytes eine RAM/EPROM-Karte notwendig. Um die Investitionen zu reduzieren, kann man sich einer der nachfolgend beschriebenen Alternativen bedienen. Beide Lösungen sind erheblich kostengünstiger als die Vektoren-RAM/EPROM-Karte, sie verlangen aber vom Junior-Computer-Besitzer ein Stück handwerkliche Arbeit. Dies wird jedoch vom finanziellen Vorteil aufgewogen.
Hier die beiden Alternativen:

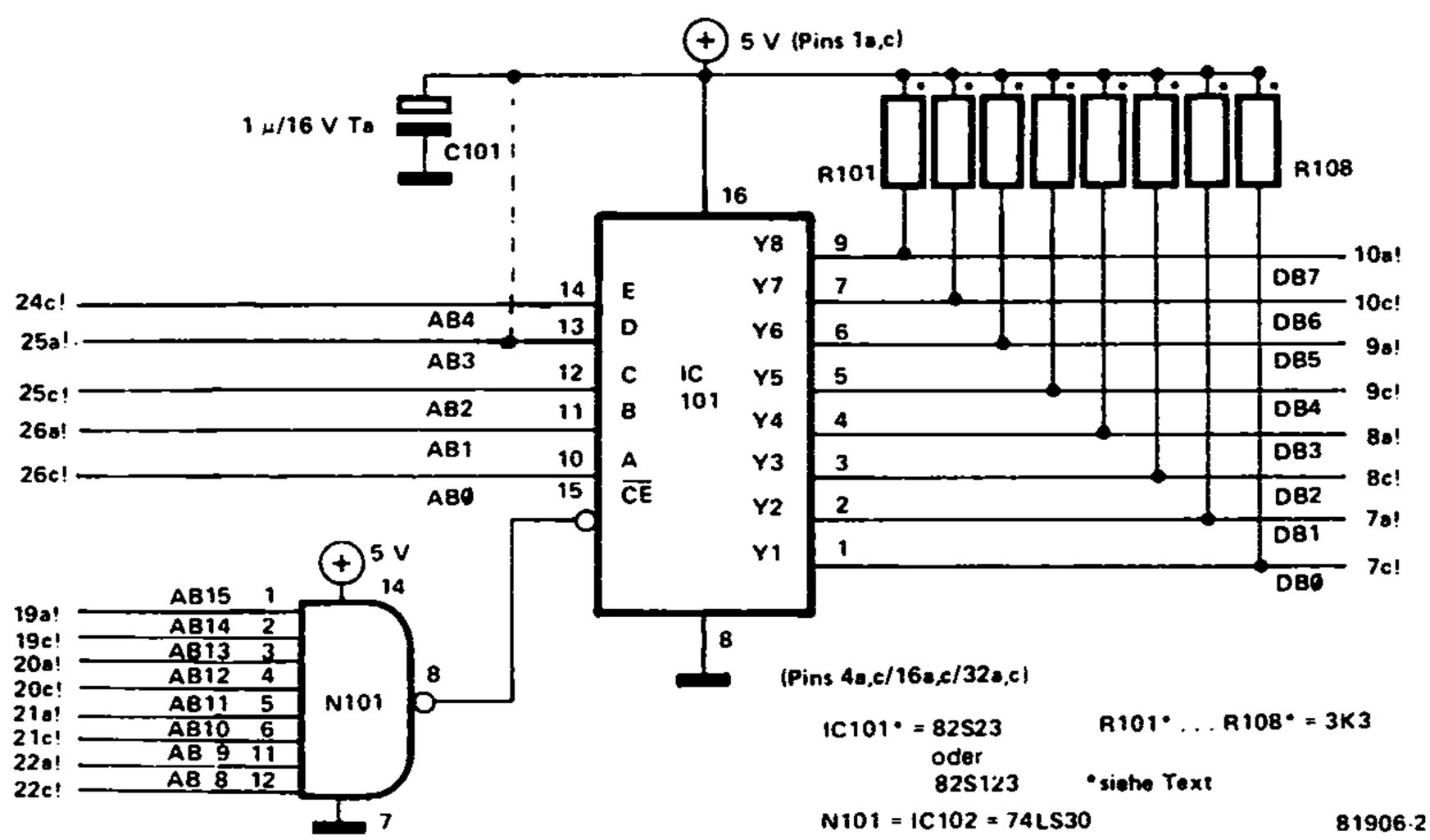
① Obwohl Busspeicherkarten angeschlossen sind, werden die Vektoren mit Hilfe zusätzlicher Hardware aus dem EPROM auf der Basiskarte gelesen.

② Auf die Buskarte wird eine Vektor-Speicherkarte gesteckt.

Lösung ①: Sprungvektoren aus dem Basiskarten-EPROM

Aus Kapitel 10 wissen wir, daß bei der Erweiterung des Junior-Computers mit Busspeicherkarten die Brücken D-EX und R-S (WITH) gelegt werden müssen, während ohne Busspeicherkarten die Brücken D-Masse und R-T ($\overline{\text{WITH}}$) vorhanden sind. Im letzten Fall liest der Prozessor die Vektoren aus dem Basiskarten-EPROM. In Bild 1 ist dies ebenfalls realisiert, obwohl Busspeicherkarten angeschlossen sind. Zur Hardware des Juniors kommen vier Gatter (zwei ICs) hinzu. Der Ausgang von N103 wird mit Punkt D und der Ausgang von N101 mit Punkt R verbunden. **Die beiden Drahtbrücken R-S und D-EX müssen entfernt werden!**
Und so arbeitet die Schaltung: Solange nicht alle vier Eingänge von N101 logisch 1 sind, ist der Ausgang dieses Gatters logisch 1. Dies ist die gleiche Situation wie bei Vorhandensein der Brücke R-S. Wenn aber alle vier Adreßleitungen A12 . . . A15 auf logisch 1 gehen, was bei den Adressen FXXX, also auch bei den Vektoradressen FFFA . . . FFFF zutrifft, ist der Ausgang von N101 logisch Ø. Dies aber ist gleichbedeutend mit der Brücke


Drahtbrücke entfällt!
IC6
EX
D
X
D
EX = 8K0
N103
N104
A15
A14
A13
A12
N101
N102
5 V
Drahtbrücke entfällt!
1µ/16 V Ta
S
(WITH)
R
E
T
(WITH)
IC 17 PROM
N101,N102 = 74LS13 = IC101
N103,N104 = ½74LS00 = IC102
81906-1


5 V (Pins 1a,c)
1 µ/16 V Ta
C101
R101
R108
16
24c!
25a!
25c!
26a!
26c!
AB4
AB3
AB2
AB1
AB0
14
13
12
11
10
15
E
D
C
B
A
CE
IC 101
Y8
Y7
Y6
Y5
Y4
Y3
Y2
Y1
9
7
6
5
4
3
2
1
10a!
10c!
9a!
9c!
8a!
8c!
7a!
7c!
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
5 V
14
19a!
19c!
20a!
20c!
21a!
21c!
22a!
22c!
AB15 1
AB14 2
AB13 3
AB12 4
AB11 5
AB10 6
AB 9 11
AB 8 12
N101
8
7
8
(Pins 4a,c/16a,c/32a,c)
IC101* = 82S23
oder
82S123
R101* . . . R108* = 3K3
*siehe Text
N101 = IC102 = 74LS30
81906-2

R-T; die erste Bedingung für das Lesen der Vektoren aus dem Basiskarten-EPROM ist damit erfüllt.
Bedingung Zwei: Wenn alle vier Adreßleitungen A12 . . . A15 logisch 1 sind, muß an Punkt D von IC6 auf der Basiskarte logisch Ø liegen. Dafür sorgen die Inverter N102 (N101 und N102 bilden zusammen ein UND-Gatter) sowie das aus N103 und N104 bestehende UND-Gatter. Obwohl das Signal $\overline{8K\emptyset}$ beim Lesen eines Vektors logisch 1 wird, hat dies nun nicht mehr eine logische 1, sondern eine Ø an Punkt D von IC6 zur Folge.
Der Aufbau der Schaltung in der Praxis und deren Anschluß an den Junior erfordert etwas handwerkliches Geschick. Am besten nimmt man hierzu eine Lochrasterplatte; sie wird in der Nähe der Verbindungsplatine zwischen Basis- und Interfacekarte montiert. Die Pinbelegungen von IC101 und IC102 sind in Kapitel 10, Bild 18 zu finden. Die Eingänge von N101 können an die ungepufferten Adreßleitungen A12 . . . A15 gelegt werden. Nicht vergessen: Brücken R-S und D-EX entfernen!
Der Adressenbereich FXXX (ein 4-K-Adressenblock) kann nicht mehr mit Buskartenspeichern belegt werden, denn bei Selektion dieses Bereichs werden mit der Schaltung in Bild 1 die Datenbuspuffer auf der Interfacekarte blockiert.
Wem der beschriebene Eingriff in die Hardware des Juniors zu schwierig erscheint, dem steht noch eine zweite, allerdings etwas teuere Lösung offen.

Lösung ②: Vektor-Speicherkarte

Man nehme ein Stück Lochrasterplatte, 10 cm breit und 5 . . . 6 cm lang, montiere auf der Breitseite einen Konnektor (männlich; siehe Bild 19a in Kapitel 10), füge die Schaltung aus Bild 2 hinzu und stecke die so entstandene Vektor-Speicherkarte auf die Buskarte. **Beim Bau der Schaltung auf richtigen Anschluß an die einzelnen Konnektorpins achten!**
Diese Schaltung arbeitet wie folgt: IC101 ist ein 32-Byte-PROM vom gleichen Typ wie das PROM auf der Interfacekarte. Sechs Speicherplätze sind mit den sechs Vektoren programmiert. Wenn Anschluß $\overline{CE}$ (Chip Enable) logisch 1 ist, sind die PROM-Ausgänge hochohmig; sie haben keinen Einfluß auf den Datenbus. Das ist dann der Fall, wenn mindestens eine der mit N101 verbundenen Adreßleitungen logisch Ø ist. Bei Adressierung von Seite FF gehen alle hier anliegenden Adreßleitungen gleichzeitig auf logisch 1. Der $\overline{CE}$-Eingang von IC101 wird dadurch logisch Ø. Das PROM setzt daraufhin den vom Prozessor adressierten Vektor auf den Datenbus. Da nur sechs Byte gelesen werden müssen, liegen die PROM-Adreßeingänge D und E an der Speisespannung. Das PROM ist wie folgt zu programmieren:

Adresse	(EDCBA; hex.)	Daten	(hex.)
ØØ . . . 19		ØØ	
1A		2F	(NMIL)
1B		1F	(NMIH)
1C		1D	(RESL)
1D		1C	(RESH)
1E		32	(IRQL)
1F		1F	(IRQH)

Obwohl die Vektoren nun nicht mehr aus dem Basiskarten-EPROM geholt werden, ist dieses natürlich nicht überflüssig geworden. Von seinem Inhalt werden zumindest noch die Resetroutine und die Sprungvektoren vom EPROM zum PIA-RAM (IRQ, BRK, NMI) benötigt.
Da das Vektor-Speicherkarten-PROM von acht Adreßleitungen selektiert wird, beträgt der Verlust an frei verfügbarem Speicherbereich hier nur 256 Byte (Seite FF). Allerdings ist Seite FF Teil eines 4-K-Bereichs, der nicht mit RAM/EPROM-Speicherkarten belegt werden kann.
Verwendet man für IC101 den EPROM-Typ 82S123, dann können die Widerstände R101 . . . R108 entfallen. Und nicht vergessen: Auf richtigen Anschluß der Leitungen mit den Konnektorpins achten!

Hexdump vom Programm Tape Monitor (TM)

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0800: 20 72 0C A5 FA 8D 70 1A A5 FB 8D 71 1A 4C 4E 08
0810: A2 00 8E 79 1A 8E 70 1A 8E 71 1A 8E 72 1A 8E 73
0820: 1A 8E 68 1A A9 06 8D 82 1A A9 1E 8D 83 1A 20 36
0830: 09 D0 FB 20 36 09 F0 FB 20 36 09 F0 F6 20 F9 1D
0840: C9 14 D0 0E 20 02 0B A2 FF EC 79 1A F0 B2 E8 4C
0850: 21 08 C9 10 D0 08 20 DF 09 A2 00 4C 21 08 C9 12
0860: D0 11 EE 68 1A A0 09 CC 68 1A D0 C2 A0 00 8C 68
0870: 1A F0 BB C9 13 D0 25 A5 E2 8D 70 1A A5 E3 8D 71
0880: 1A A5 E8 8D 72 1A A5 E9 8D 73 1A 20 DF 09 20 D3
0890: 1E A9 1E 8D 83 1A A2 FF 9A 4C CA 1C C9 11 D0 03
08A0: 4C B5 1C 06 F9 06 F9 06 F9 06 F9 05 F9 85 F9 A0
08B0: 00 CC 68 1A D0 06 8D 79 1A 4C 2E 08 C8 CC 68 1A
08C0: D0 06 8D 71 1A 4C 2E 08 C8 CC 68 1A D0 06 8D 70
08D0: 1A 4C 2E 08 C8 CC 68 1A D0 06 8D 73 1A 4C 2E 08
08E0: C8 CC 68 1A D0 06 8D 72 1A 4C 2E 08 C8 CC 68 1A
08F0: D0 05 85 E3 4C 2E 08 C8 CC 68 1A D0 05 85 E2 4C
0900: 2E 08 C8 CC 68 1A D0 05 85 E5 4C 2E 08 C8 CC 68
0910: 1A D0 05 85 E4 4C 2E 08 4C 10 08 B9 BB 09 8D 80
0920: 1A 8E 82 1A 98 A0 7F 88 D0 FD A8 E8 E8 C8 E0 10
0930: D0 E9 20 A7 1D 60 A9 7F 8D 81 1A A2 08 A0 00 CC
0940: 68 1A D0 09 AD 79 1A 85 F9 20 1B 09 60 C8 CC 68
0950: 1A D0 09 AD 71 1A 85 F9 A0 04 D0 ED C8 CC 68 1A
0960: D0 09 AD 70 1A 85 F9 A0 08 D0 DE C8 CC 68 1A D0
0970: 09 AD 73 1A 85 F9 A0 0C D0 CF C8 CC 68 1A D0 09
0980: AD 72 1A 85 F9 A0 10 D0 C0 C8 CC 68 1A D0 08 A5
0990: E3 85 F9 A0 14 D0 B2 C8 CC 68 1A D0 08 A5 E2 85
09A0: F9 A0 18 D0 A4 C8 CC 68 1A D0 08 A5 E5 85 F9 A0
09B0: 1C D0 96 A5 E4 85 F9 A0 20 D0 8E 66 21 7F 7F 52
09C0: 08 09 7F 52 08 47 7F 06 08 09 7F 06 08 47 7F 03
09D0: 06 42 09 03 06 42 47 06 2B 21 09 06 2B 21 47 A9
09E0: 7D 8D 6C 1A A9 C3 8D 6D 1A A9 03 8D 76 1A A9 02
09F0: 8D 77 1A A9 47 A2 FF 8D 82 1A 8D 78 1A 8E 83 1A
```

```
0A00: A9 00 A2 7F 8D 80 1A 8E 81 1A 8D 6E 1A 8D 6F 1A
0A10: AD 70 1A 85 FA AD 71 1A 85 FB A2 FF 8E 74 1A A9
0A20: 16 20 A3 0A CE 74 1A D0 F6 A9 2A 20 A3 0A AD 79
0A30: 1A 20 8B 0A AD 70 1A 20 7A 0A AD 71 1A 20 7A 0A
0A40: A5 FB CD 73 1A D0 23 A5 FA CD 72 1A D0 1C A9 2F
0A50: 20 A3 0A AD 6E 1A 20 8B 0A AD 6F 1A 20 8B 0A A9
0A60: 04 20 A3 0A A9 04 20 A3 0A 60 A0 00 B1 FA 20 7A
0A70: 0A E6 FA D0 CB E6 FB 4C 40 0A A8 18 6D 6E 1A 8D
0A80: 6E 1A AD 6F 1A 69 00 8D 6F 1A 98 A8 4A 4A 4A 4A
0A90: 20 9A 0A 98 29 0F 20 9A 0A 60 C9 0A 18 30 02 69
0AA0: 07 69 30 A2 08 8E 75 1A 4A 48 90 0C 20 C8 0A 20
0AB0: E5 0A 20 E5 0A 4C C1 0A 20 C8 0A 20 C8 0A 20 E5
0AC0: 0A 68 CE 75 1A D0 E1 60 AE 76 1A 2C D5 1A 10 FB
0AD0: AD 6C 1A 8D F4 1A AD 78 1A 49 80 8D 82 1A 8D 78
0AE0: 1A CA D0 E7 60 AE 77 1A 2C D5 1A 10 FB AD 6D 1A
0AF0: 8D F4 1A AD 78 1A 49 80 8D 82 1A 8D 78 1A CA D0
0B00: E7 60 A9 32 8D 82 1A 8D 78 1A A9 7E 8D 83 1A A9
0B10: 7F 8D 81 1A A9 00 8D 6E 1A 8D 6F 1A A9 FF 8D 6B
0B20: 1A 20 C2 0B 6E 6B 1A AD 6B 1A 20 E8 0B C9 16 D0
0B30: F0 A0 0A 8C 69 1A 20 36 0C 20 5D 0C C9 16 D0 DC
0B40: CE 69 1A D0 F1 20 36 0C 20 5D 0C C9 2A F0 06 C9
0B50: 16 F0 F2 D0 AD 20 5D 0C 20 F3 0B CD 79 1A D0 3E
0B60: 20 F3 0B 20 4B 0C 85 FA 20 F3 0B 20 4B 0C 85 FB
0B70: 20 F3 0B 30 8D F0 13 20 4B 0C A0 00 91 FA E6 FA
0B80: D0 02 E6 FB 20 64 0C 4C 70 0B 20 F3 0B CD 6E 1A
0B90: D0 09 20 F3 0B CD 6F 1A D0 01 60 4C 02 0B AD 79
0BA0: 1A C9 00 F0 BB C9 FF D0 F2 20 F3 0B 20 4B 0C 20
0BB0: F3 0B 20 4B 0C AD 70 1A 85 FA AD 71 1A 85 FB 4C
0BC0: 70 0B 2C 82 1A 10 FB AD D4 1A A0 FF 8C F6 1A A0
0BD0: 14 88 D0 FD 2C 82 1A 30 FB 38 ED D4 1A A0 FF 8C
0BE0: F6 1A A0 07 88 D0 FD 60 48 A9 36 8D 80 1A 68 20
0BF0: 64 0C 60 20 36 0C C9 2F D0 01 60 20 19 0C 30 FA
0C00: 0A 0A 0A 0A 8D 6A 1A 20 36 0C C9 2F F0 EC 20 19
0C10: 0C 30 E7 0D 6A 1A A0 01 60 C9 30 30 0C C9 3A 30
0C20: 0B C9 14 30 04 C9 47 30 03 A0 FF 60 C9 40 30 03
0C30: 18 69 09 29 0F 60 A2 08 20 C2 0B 6E 6B 1A CA D0
0C40: F7 2E 6B 1A 4E 6B 1A AD 6B 1A 60 48 18 6D 6E 1A
0C50: 8D 6E 1A AD 6F 1A 69 00 8D 6F 1A 68 60 48 49 7F
0C60: 8D 80 1A 68 48 AD 78 1A 49 02 8D 82 1A 8D 78 1A
0C70: 68 60 38 A5 FA E9 01 85 FA A5 FB E9 00 85 FB 60
```

Anhang 5

Hexdump vom Programm Printer Monitor (PM)

```
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
1000: D8 78 A9 67 8D 82 1A A9 00 8D 80 1A A2 FE 8E 5A
1010: 1A E8 8E 5B 1A 9A 86 F2 A9 7F 8D 81 1A 8D 83 1A
1020: A2 02 8E 59 1A A9 5F 8D 7C 1A A9 10 8D 7D 1A A9
1030: CF 8D 7A 1A A9 14 8D 7B 1A 2C 80 1A 30 FB 20 E0
1040: 12 4E 5F 1A 6E 5E 1A AD 5E 1A 8D 5C 1A AD 5F 1A
1050: 8D 5D 1A A2 08 20 03 13 20 BE 12 C9 7F D0 A1 20
1060: 46 12 20 E8 11 A2 FF 9A 86 F2 20 59 12 20 68 12
1070: 20 AE 12 C9 2B D0 09 20 13 12 20 F8 11 4C 6A 10
1080: C9 2D D0 09 20 1A 12 20 F8 11 4C 6A 10 C9 20 D0
1090: 0E A5 F8 85 FA A5 F9 85 FB 20 F8 11 4C 6A 10 C9
10A0: 2E D0 0F A5 F8 A0 00 91 FA 20 13 12 20 F8 11 4C
10B0: 6A 10 C9 52 D0 13 A6 F2 9A A5 FB 48 A5 FA 48 A5
10C0: F1 48 A6 F5 A4 F4 A5 F3 40 C9 4C D0 52 A0 14 20
10D0: D6 11 A5 F3 20 8F 12 A0 1A 20 D6 11 A5 F4 20 8F
10E0: 12 A0 20 20 D6 11 A5 F5 20 8F 12 A0 26 20 D6 11
10F0: A5 F0 20 8F 12 A5 EF 20 8F 12 A0 2C 20 D6 11 A9
1100: 01 20 8F 12 A5 F2 20 8F 12 A0 32 20 D6 11 20 28
1110: 12 A0 38 20 D6 11 20 F3 11 4C 6A 10 4C B0 11 C9
1120: 50 D0 0E A5 EF 85 FA A5 F0 85 FB 20 F8 11 4C 6A
1130: 10 C9 4D D0 E7 A0 52 20 D6 11 20 87 13 10 03 4C
1140: 5F 10 38 AD 65 1A ED 63 1A AD 66 1A ED 64 1A 90
1150: EE 20 E8 11 A2 06 20 F3 11 CA D0 FA A0 00 98 20
1160: 9B 12 20 F3 11 20 F3 11 C8 C0 10 D0 F1 AD 63 1A
1170: 85 FA AD 64 1A 85 FB 20 E8 11 A2 10 A5 FB 20 8F
1180: 12 A5 FA 20 8F 12 A0 17 20 D9 11 38 AD 65 1A E5
1190: FA AD 66 1A E5 FB B0 06 20 E8 11 4C 5F 10 A0 00
11A0: B1 FA 20 8F 12 20 F3 11 20 13 12 CA D0 DD F0 C7
11B0: C9 47 D0 0B 20 8A 14 30 03 4C 6A 10 4C 5F 10 C9
11C0: 53 D0 08 20 3B 14 30 F4 4C 6A 10 20 6F 12 D0 03
11D0: 4C 70 10 4C 6A 10 20 E8 11 B9 BD 13 C9 03 F0 07
11E0: 20 34 13 C8 4C D9 11 60 A9 0D 20 34 13 A9 0A 20
11F0: 34 13 60 A9 20 4C EF 11 20 E8 11 A5 FB 20 8F 12
1200: A5 FA 20 8F 12 20 F3 11 A0 00 B1 FA 20 8F 12 20
1210: F3 11 60 E6 FA D0 02 E6 FB 60 38 A5 FA E9 01 85
1220: FA A5 FB E9 00 85 FB 60 A5 F1 8D 67 1A A2 08 0E
1230: 67 1A 90 09 A9 01 20 9B 12 CA D0 F3 60 A9 00 20
```

```
1240: 9B 12 CA D0 EA 60 A0 00 20 D6 11 20 E8 11 60 A0
1250: 07 4C 48 12 A0 0E 4C 48 12 A0 00 8C 63 1A 8C 64
1260: 1A 8C 65 1A 8C 66 1A 60 A0 00 84 F8 84 F9 60 20
1270: 1E 14 30 10 A2 04 06 F8 26 F9 CA D0 F9 05 F8 85
1280: F8 A0 00 60 A0 46 20 D6 11 20 E8 11 A0 FF 60 48
1290: 4A 4A 4A 4A 20 A4 12 20 34 13 68 29 0F 20 A4 12
12A0: 20 34 13 60 C9 0A 18 30 02 69 07 69 30 60 2C 80
12B0: 1A 30 FB 8E 61 1A A2 08 20 03 13 20 12 13 2C 80
12C0: 1A 10 0A 38 6E 62 1A CA D0 F1 4C D4 12 18 6E 62
12D0: 1A CA D0 E7 20 12 13 AD 62 1A 29 7F AE 61 1A 60
12E0: 18 AD 5A 1A 69 01 8D 5A 1A AD 5B 1A 69 00 8D 5B
12F0: 1A 2C 80 1A 10 EA AD 5A 1A 8D 5E 1A AD 5B 1A 8D
1300: 5F 1A 60 AD 5C 1A 8D 5E 1A AD 5D 1A 8D 5F 1A 4C
1310: 1E 13 AD 5A 1A 8D 5E 1A AD 5B 1A 8D 5F 1A 38 AD
1320: 5E 1A E9 01 8D 5E 1A AD 5F 1A E9 00 8D 5F 1A EA
1330: EA B0 EB 60 8E 60 1A 8D 62 1A AD 82 1A 29 FE 8D
1340: 82 1A 20 12 13 A2 07 4E 62 1A 90 30 AD 82 1A 09
1350: 01 8D 82 1A 20 12 13 CA D0 ED AE 59 1A AD 82 1A
1360: 09 01 8D 82 1A 20 12 13 CA D0 F2 2C 80 1A 10 04
1370: AE 60 1A 60 2C 80 1A 10 FB 6C 7C 1A AD 82 1A 29
1380: FE 8D 82 1A 4C 54 13 20 AE 12 C9 2C F0 07 20 6F
1390: 12 30 29 F0 F2 A5 F8 8D 63 1A A5 F9 8D 64 1A 20
13A0: 68 12 20 AE 12 C9 0D F0 07 20 6F 12 30 0E F0 F2
13B0: A5 F9 8D 66 1A A5 F8 8D 65 1A A0 00 60 4A 55 4E
13C0: 49 4F 52 03 45 44 49 54 4F 52 03 41 53 53 45 4D
13D0: 03 41 43 43 3A 20 03 59 20 20 3A 20 03 58 20 20
13E0: 3A 20 03 50 43 20 3A 20 03 53 50 20 3A 20 03 50
13F0: 52 20 3A 20 03 20 20 20 20 20 4E 56 20 42 44 49
1400: 5A 43 03 57 48 41 54 3F 03 52 45 41 44 59 03 48
1410: 45 58 44 55 4D 50 3A 20 03 53 41 3A 20 03 C9 30
1420: 30 0C C9 3A 30 0B C9 41 30 04 C9 47 30 03 A0 FF
1430: 60 C9 40 30 03 18 69 09 29 0F 60 20 AE 12 C9 2C
1440: F0 07 20 6F 12 30 3F F0 F2 A5 F8 8D 79 1A C9 00
1450: F0 35 C9 FF F0 31 20 68 12 20 87 13 30 28 AD 63
1460: 1A 8D 70 1A AD 64 1A 8D 71 1A AD 65 1A 8D 72 1A
1470: AD 66 1A 8D 73 1A 20 DF 09 20 BC 14 A0 4C 20 D6
1480: 11 20 E8 11 A0 00 60 A0 FF 60 20 A2 13 30 17 8D
1490: 79 1A C9 FF F0 11 20 02 0B 20 BC 14 A0 4C 20 D6
14A0: 11 20 E8 11 A0 00 60 A0 5C 20 D6 11 20 EB 14 30
14B0: F5 8D 70 1A A5 F9 8D 71 1A 4C 96 14 A9 67 8D 82
14C0: 1A A9 00 8D 80 1A A9 7F 8D 81 1A 8D 83 1A 60 85
14D0: F3 68 85 F1 68 85 EF 85 FA 68 85 F0 85 FB 84 F4
14E0: 86 F5 BA 86 F2 20 F8 11 4C 6A 10 A9 00 85 F8 85
14F0: F9 4C A2 13
```

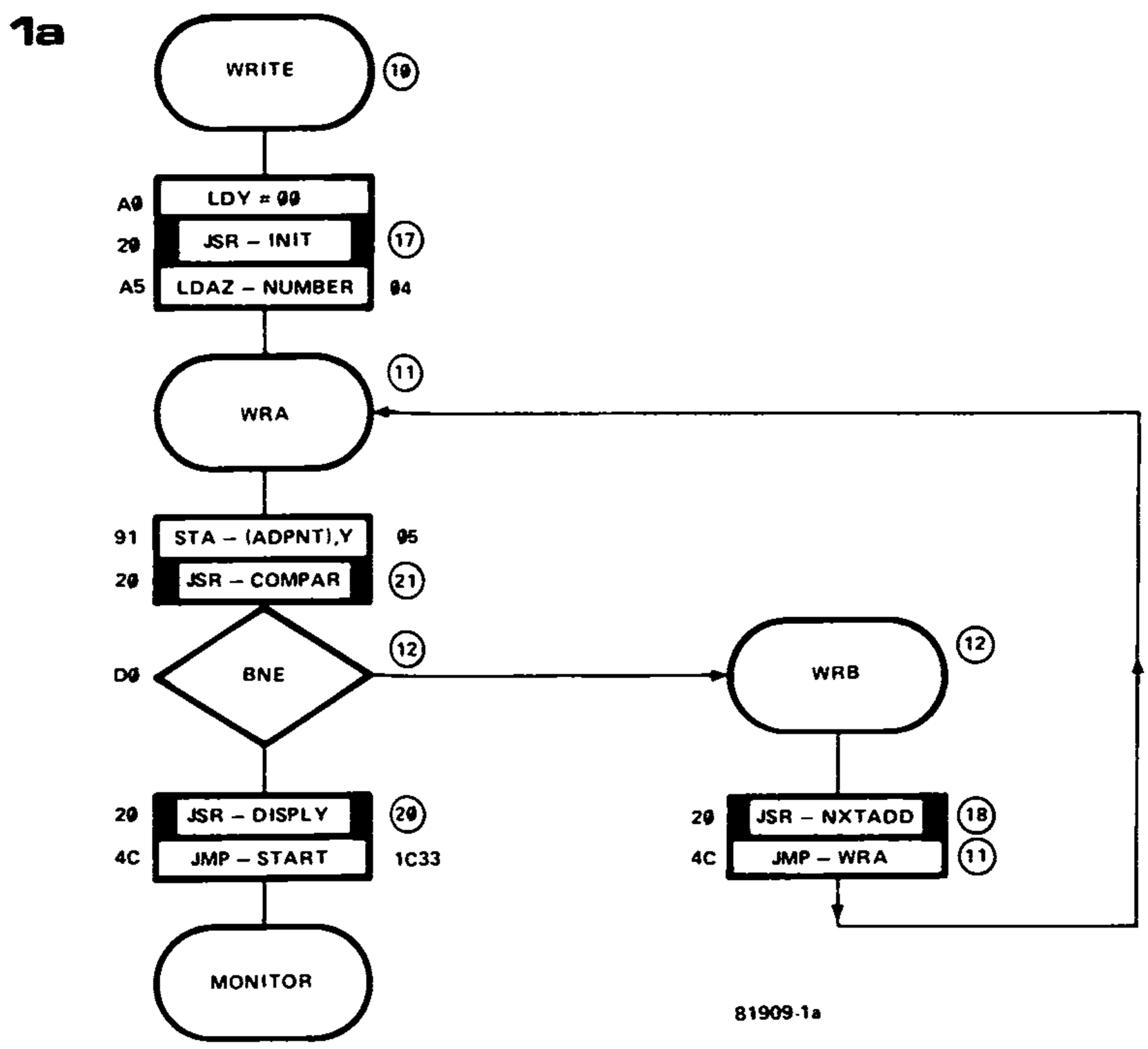
Anhang 6

RAM-Testprogramm

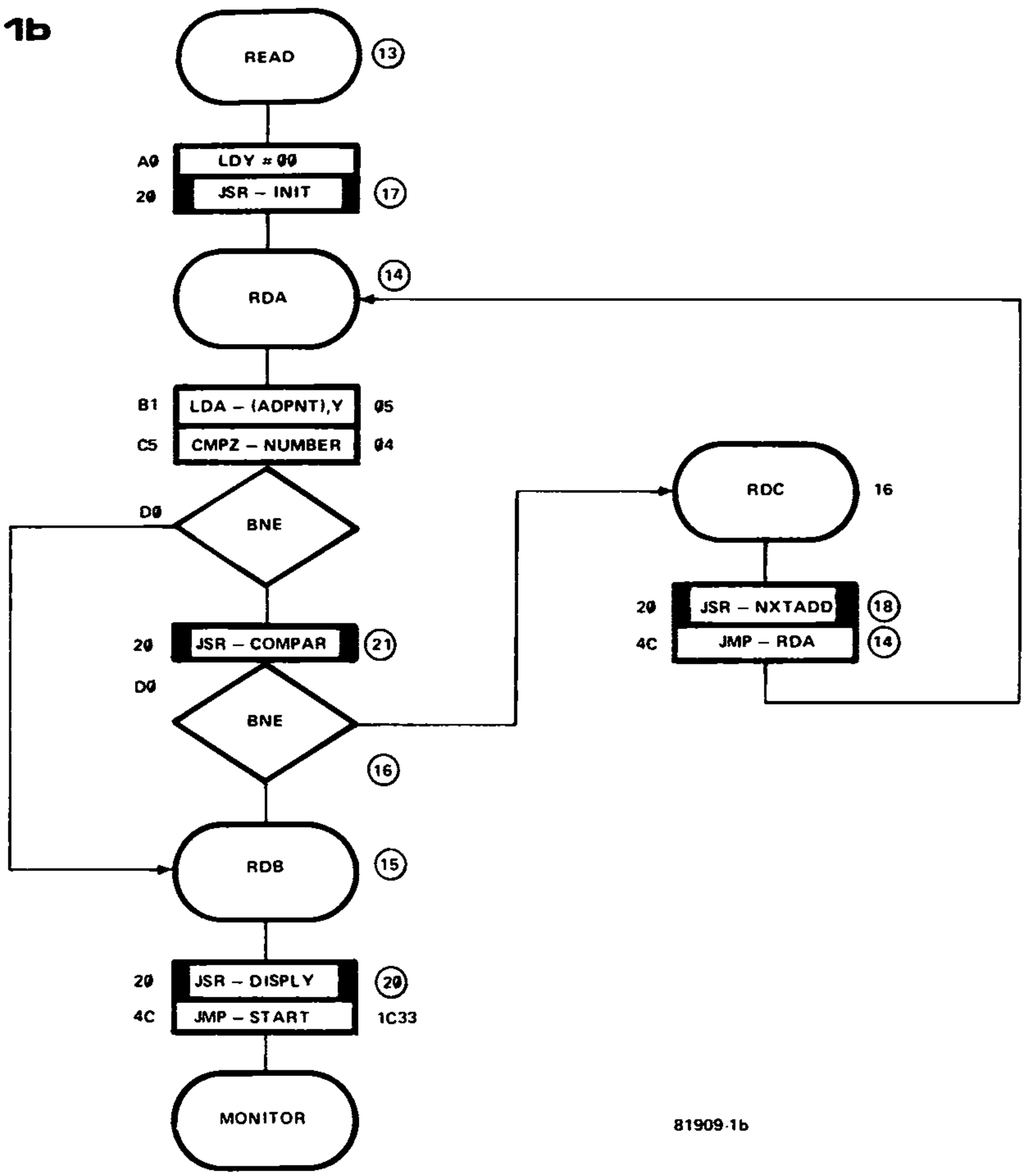
Schreib- und Lesetest

Zur Erweiterung des Junior-Computers gehört normalerweise auch die Ausdehnung des RAM-Speicherbereichs. Unabhängig davon, ob die zusätzlichen RAMs auf der Interfacekarte, auf RAM/EPROM-Karten oder auf dynamischen RAM-Karten sitzen, ist mit diesem Testprogramm eine Kontrolle auf korrektes Schreiben und Lesen möglich.
Bild 1 zeigt die "Einzelteile" des Testprogramms. Es wird mit dem Editor und dem Assembler in das RAM auf der Basiskarte gesetzt (Seiten Ø2 . . . Ø3 oder ein Teil von Seite ØØ). Innerhalb bestimmter Grenzen sind

1a

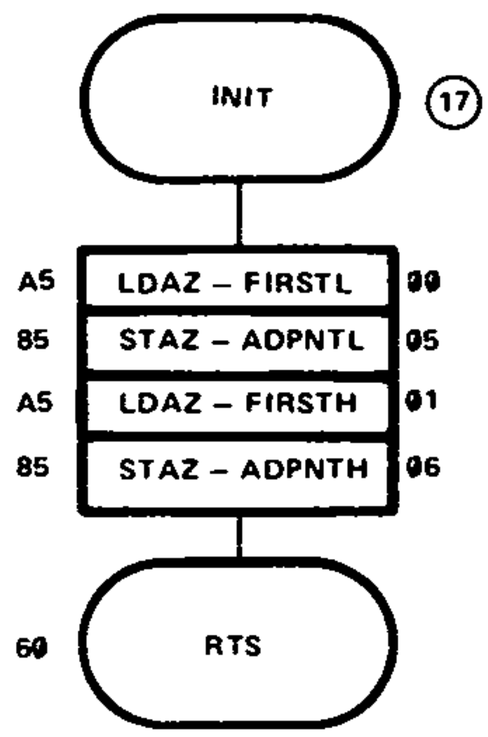


1b



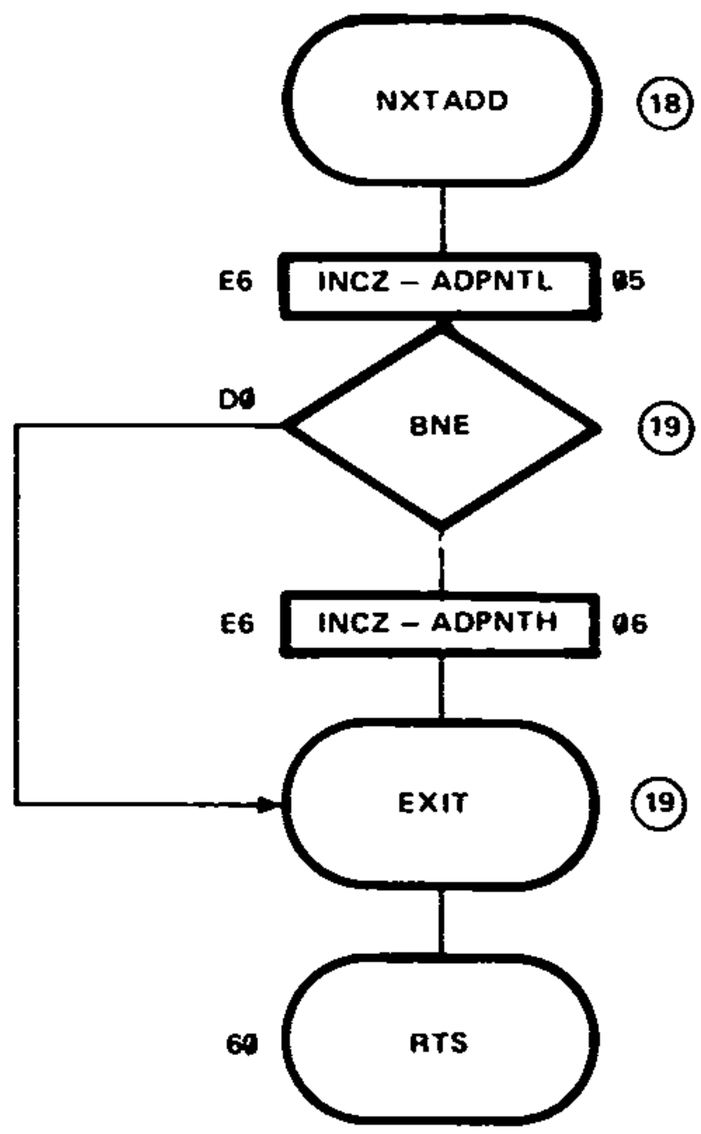
die Startadressen frei wählbar; darauf deuten schon die Label in Bild 1 hin. Da das Testprogramm in das RAM der Basiskarte geladen wird, muß dieser Teil des erweiterten Junior-Computers mit Sicherheit einwandfrei arbeiten. Das Testprogramm enthält eine Schreibroutine (Bild 1a), eine Leseroutine (Bild 1b) und vier Subroutinen. Während des Schreibens werden alle Speicherplätze von Adresse FIRST bis einschließlich Adresse LAST mit Daten geladen, die in Speicherplatz NUMBER stehen. Nachdem dies geschehen ist, erscheint Adresse LAST mit den Daten von NUMBER auf den Displays. Anschließend werden alle Speicherzellen des zu testenden Bereichs mit dem Leseprogramm READ (Bild 1b) gelesen und mit dem Inhalt von NUMBER verglichen. Stimmt der Inhalt einer Speicherzelle nicht mit NUMBER überein, dann springt das Programm zum Monitor zurück. Auf den Displays erscheint die Adresse dieser Speicherzelle zusammen mit den dort stehenden Daten.

1c



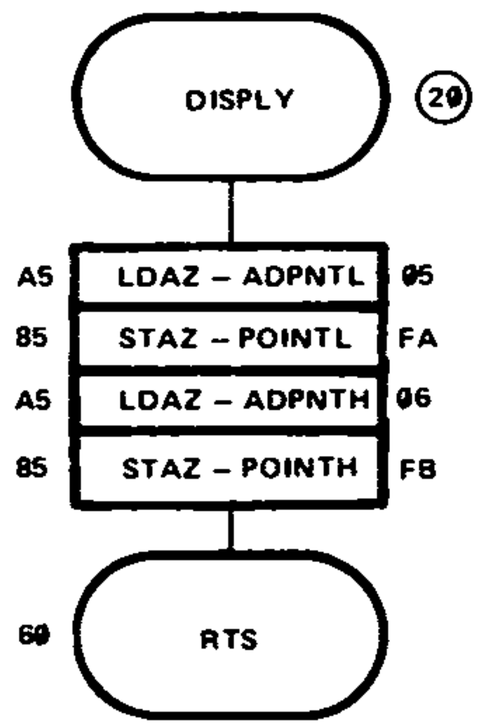
81909-1c

1d



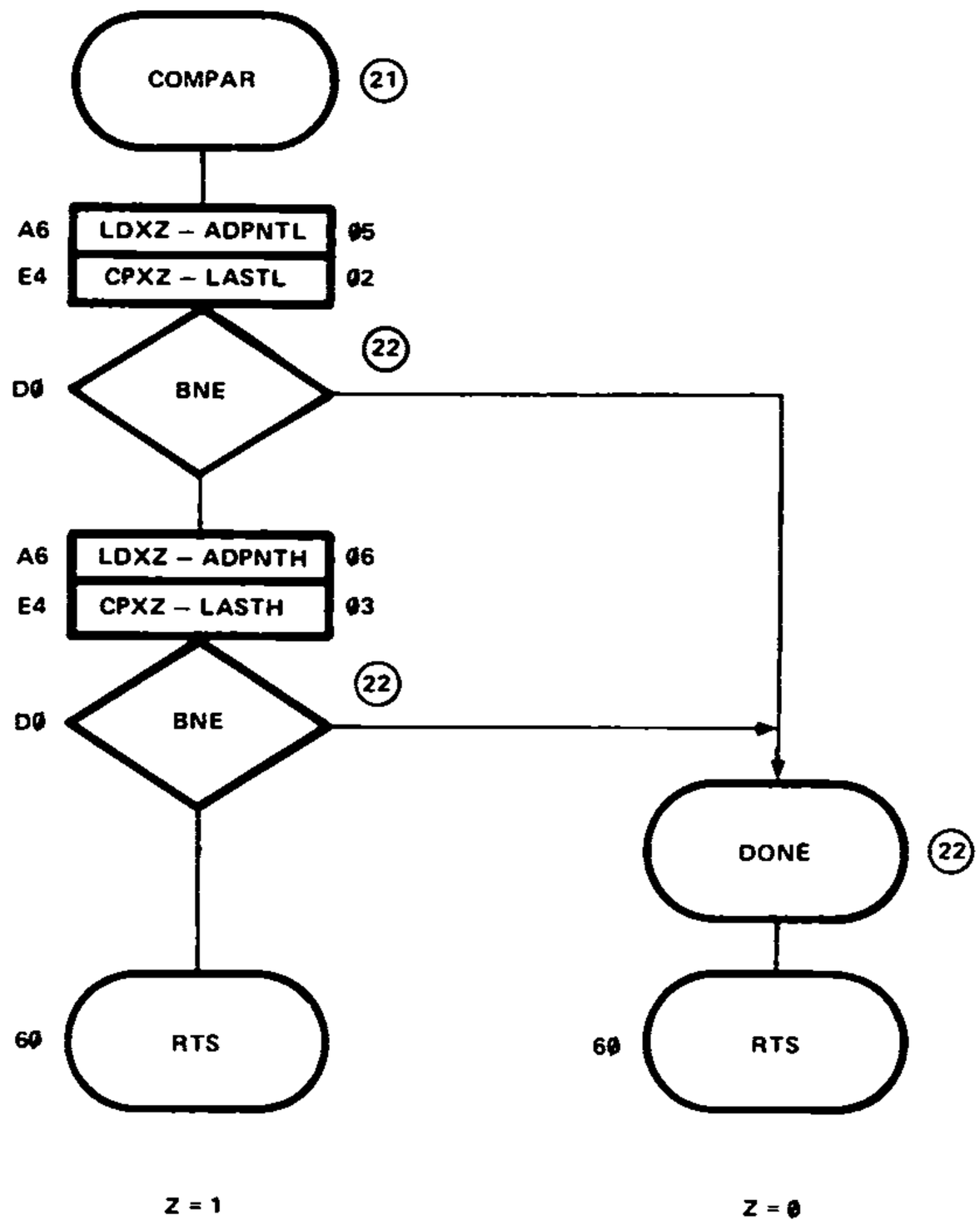
81909-1d

1e



81909-1e

1f



Gebrauchsanweisung

① Das Programm aus Bild 1 wird mit Hilfe des Editors und des Assemblers in das Basiskarten-RAM gesetzt; die Startadressen von WRITE und READ werden notiert.

② Folgende Adressen auf Seite 00 werden vom Programm belegt:

0000 FIRSTL 0001 FIRSTH	erste Adresse des zu testenden RAM-Bereichs
0002 LASTL 0003 LASTH	letzte Adresse des zu testenden RAM-Bereichs
0004 NUMBER	Testdaten
0005 ADPNTL 0006 ADPNTH	Schreib-/Lese-Adreßpointer

FIRST und LAST sowie NUMBER werden eingegeben. Als Testdaten sind 00 und FF weniger geeignet.

③ Start des Schreibprogramms WRITE. Nach Ende des Schreibprogramms erscheint die letzte Adresse mit den Testdaten auf den Displays.

④ Start des Leseprogramms READ. Wenn nach Ende des Leseprogramms die Adresse LAST auf den Displays sichtbar ist, wurden alle zwischen FIRST und LAST liegenden Speicherzellen einwandfrei geschrieben und gelesen. Erscheint eine niedrigere Adresse als LAST mit falschen (von NUMBER abweichenden) Daten, dann ist der untersuchte Speicherbereich nur bis zu der vorangegangenen Adresse in Ordnung. Die Adresse des fehlerhaften Speicherplatzes wird notiert.

⑤ Tritt beim ersten Test kein Fehler auf, so wird der Test mit anderen Testdaten (Inhalt von NUMBER) wiederholt. Es können nämlich in einer nicht schreibfähigen Speicherzelle durch Zufall bereits Daten stehen, die mit dem Inhalt von NUMBER identisch sind. Nach Einschalten der Speisespannung sind die Speicherzellen bekanntlich mit willkürlichen Daten geladen.

⑥ Wenn man auf einen fehlerhaften Speicherplatz stößt, können auch noch die nachfolgenden Zellen überprüft werden. Dazu wird LAST gleich der Adresse gesetzt, die um eine höher als die gefundene Adresse ist; das Programm READ wird zum zweiten Mal getestet. Dies läßt sich so oft wie nötig wiederholen.

Eine Gewissensfrage: Gesetzt den Fall, das Testprogramm stoppt plötzlich bei einer Reihe aufeinanderfolgender Adressen, deren Inhalt nicht mit NUMBER übereinstimmt. Ist dieser Bereich tatsächlich mit einem RAM und nicht mit einem EPROM, mit dem PIA oder dem VIA belegt? Dann wäre sicherlich kein Anlaß zur Beunruhigung gegeben. Das Testprogramm gibt nur über den Zustand von Adressenbereichen Auskunft, deren Inhalt vom Prozessor schreib- und lesbar ist.

⑦ Handelt es sich tatsächlich um einen RAM-Bereich, dann ist die betreffende Hardware einer eingehenden Kontrolle zu unterziehen. Ist die Adressenselektion und -dekodierung in Ordnung? Liegen die Drahtbrücken richtig? Haben sie einwandfreien Kontakt?

Bestimmte Teilbereiche der Seiten ØØ und Ø1 können nicht mit dem Programm getestet werden! Das Programm würde in diesem Fall Speicherplätze überschreiben, die während des Programmablaufs für interne Zwecke benötigt werden.

Nachdem ein Fehler gefunden wurde, kann man wie unter ⑥ angegeben nach weiteren Fehlern suchen. Statt der Anpassung von FIRST und der Eingabe der Startadresse von READ läßt sich dazu auch die Interruptroutine aus Bild 2 verwenden, die durch Drücken der Taste ST aufgerufen wird. Der Prozessor setzt dann das Leseprogramm beginnend mit dem Label RDC fort. Das Programm stoppt bei jeder Adresse mit falschem

2

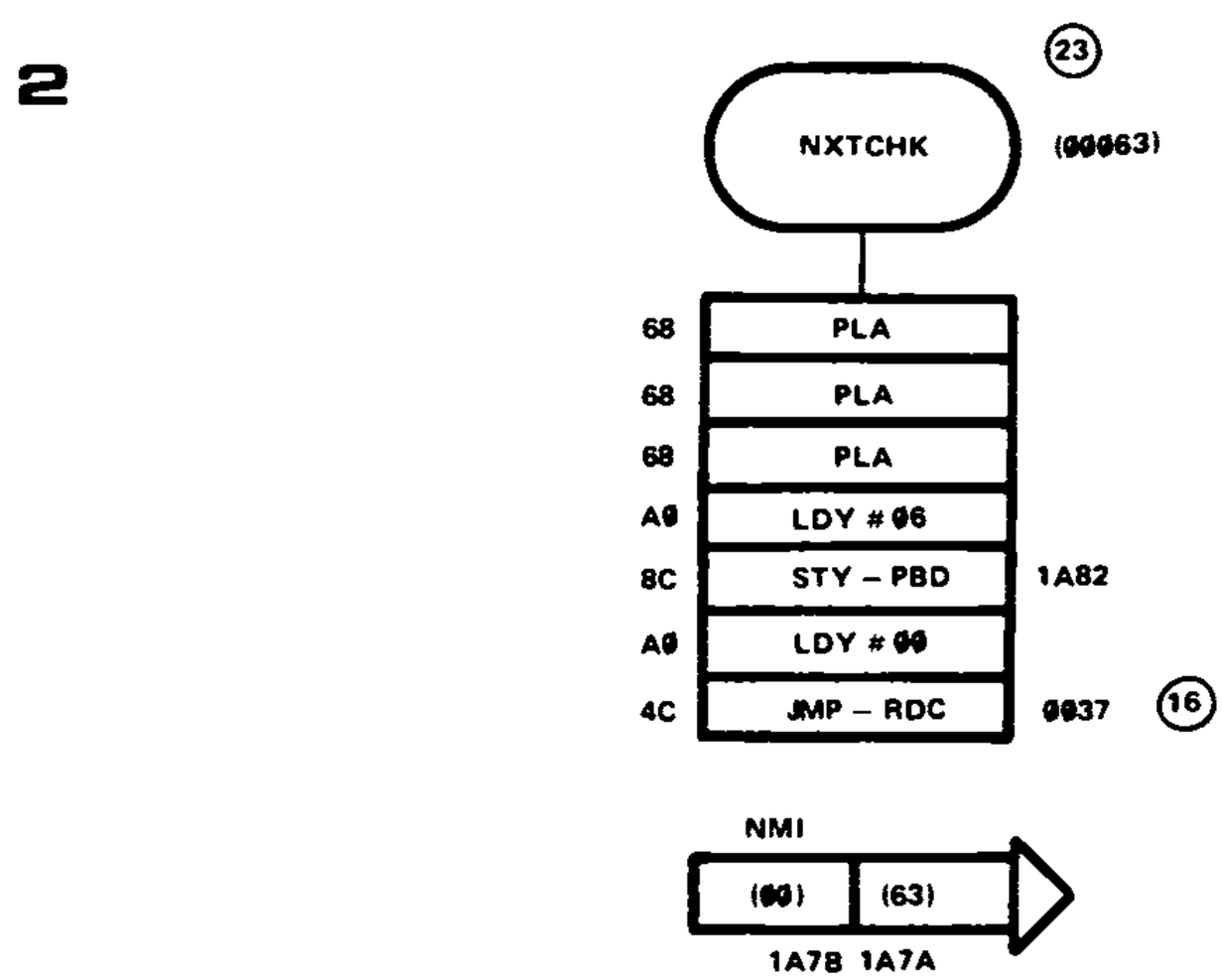


Tabelle 1

WRITE	0007	A0 00	LDY # 00
	0009	20 3D 00	JSR-INIT
	000C	A5 04	LDAZ-NUMBER
WRA	000E	91 05	STA-(ADPNT),Y
	0010	20 56 00	JSR-COMPAR
	0013	D0 06	BNE WRB
	0015	20 4D 00	JSR-DISPLY
	0018	4C 33 1C	JMP-START
WRB	001B	20 46 00	JSR-NXTADD
	001E	4C 0E 00	JMP-WRA
READ	0021	A0 00	LDY # 00
	0023	20 3D 00	JSR-INIT
RDA	0026	B1 05	LDA-(ADPNT),Y
	0028	C5 04	CMPZ-NUMBER
	002A	D0 05	BNE RDB
	002C	20 56 00	JSR-COMPAR
	002F	D0 06	BNE RDC
RDB	0031	20 4D 00	JSR-DISPLY
	0034	4C 33 1C	JMP-START
RDC	0037	20 46 00	JSR-NXTADD
	003A	4C 26 00	JMP-RDA
INIT	003D	A6 00	LDAZ-FIRSTL
	003F	85 05	STAZ-ADPNTL
	0041	A5 01	LDAZ-FIRSTH
	0043	85 06	STAZ-ADPNTH
	0045	60	RTS
NXTADD	0046	E6 05	INCZ-ADPNTL
	0048	D0 02	BNE EXIT
	004A	E6 06	INCZ-ADPNTH
EXIT	004C	60	RTS
DISPLY	004D	A5 05	LDAZ-ADPNTL
	004F	85 FA	STAZ-POINTL
	0051	A5 06	LDAZ-ADPNTH
	0053	85 FB	STAZ-POINTH
	0055	60	RTS
COMPAR	0056	A6 05	LDXZ-ADPNTL
	0058	E4 02	CPXZ-LASTL
	005A	D0 06	BNE DONE
	005C	A6 06	LDXZ-ADPNTH
	005E	E4 03	CPXZ-LASTH
	0060	D0 00	BNE DONE
DONE	0062	60	RTS
NXTCHK	0063	68	PLA
	0064	68	PLA
	0065	68	PLA
	0066	A0 06	LDY # 06
	0068	8C 82 1A	STY-PBD
	006B	A0 00	LDY # 00
	006D	4C 37 00	JMP-RDC

Inhalt. Die Adresse wird notiert, anschließend wird ST gedrückt usw. Hierbei kann allerdings Adresse LAST überschritten werden, was natürlich zwangsläufig zu Fehlschlüssen führt. Deshalb: Test bei Adresse LAST abbrechen!

Eine andere Methode

Die Arbeitsweise des RAM läßt sich auch mit Hilfe des Monitors überprüfen. Im Datenmodus wird jedesmal nach Drücken einer numerischen Taste die Instruktion STA-(POINTL), Y ausgeführt; während der periodischen Steuerung der Displays (Multiplexing) erledigt der Prozessor regelmäßig die Instruktion LDA-(POINTL), Y. Demnach wird auch hier geschrieben und gelesen.
Man gibt über das Keybaord folgendes ein:

Tasten	Displays
AD X X X X	XXXX YY
DA ZZ	XXXX ZZ

XXXX ist eine zu dem Bereich gehörende Adresse, die getestet werden soll. Die Daten ZZ werden so gewählt, daß sie sich von YY möglichst deutlich unterscheiden. Erscheinen nach der Eingabe nicht die Daten ZZ auf der Anzeige, dann liegt ein Fehler vor. Auf diese Weise kann man in "Handarbeit" jeden beliebigen Bereich durchlaufen: Man drückt Taste + und gibt von der Anzeige abweichende Daten ein.

Nachtrag: Eine assemblierte Version des Testprogramms aus den Bildern 1 und 2 (einschließlich der ST-Routine NXTCHK) ist in Tabelle 1 angegeben. Das Programm schließt sich auf Seite ØØ an die Plätze der Parameter bzw. des variabelen Adreßpointers an (ØØØØ . . . ØØØ6). Die Startadresse von WRITE ist ØØØ7, die von READ ist ØØ21. Die ST-Routine startet mit ØØ63; auf diese Adresse ist der NMI-Sprungvektor zu richten. Natürlich muß Adresse FIRST höher als ØØ6F liegen!

Anhang 7

ASCII-Codetabelle

Charakter	ASCII-Code (Hexadezimal)	Charakter	ASCII-Code (Hexadezimal)
Ø	3Ø	W	57
1	31	X	58
2	32	Y	59
3	33	Z	5A
4	34	SYN	16
5	35	Zwischenraum	2Ø
6	36	.	2E
7	37	(	28
8	38	+	2B
9	39	i	7C
A	41	&	26
B	42	!	21
C	43	$	24
D	44	*	2A
E	45	)	29
F	46	;	3B
G	47	– oder ~	7E
H	48	–	2D
I	49	/	2F
J	4A	,	2C
K	4B	%	25
L	4C	– oder ←	5F
M	4D	>	3E
N	4E	?	3F
O	4F	:	3A
P	5Ø	#	23
Q	51	@	4Ø
R	52	'	27
S	53	=	3D
T	54	"	22
U	55	<	3C
V	56	EOT	64

Sachwortverzeichnis

Adreßdekodierung 16
Adresse 83
  letzte - LA 83
  End - EA 83
ASCII
  -Kode 176
  -Tastatur . . . . . 121, 132 ff.
Aufsteckplatine 36
Aufzeichnungsgeschwindigkeit . . . 28

Bandkassette 81
Baud 28
Baudrate 28, 96
Bus 8
Busspeicherkarte 68

CHSBRD 171

Datenarchiv 9
Datenblock 82, 84
Datenblockbeginnzeichen 83
Datenblockendzeichen 84
Datenbuspuffer 11
Datenrücktransport 90
Datensende-Endzeichen EOT 87
Datensendung 82
Datenübertragung 122
DECAR 160
D-Konnektor 29, 137
Dropout 85

Einlesen des Programms 91
Elekterminal 121

Fehlermeldungen 163
FINAL 176
Frequenzfolger 27
FSK 27

Jitter 91

Kassetten-Hardware 25
Komparator 28
Konnektor 29
Kontrollbytes CHKL, CHKH 85

Magnetische Speichererweiterung . 80
Manipulation der Daten 94

NUMBERS . . . . . 150ff.

Parameter 84
Paritätsbit 127
Phase Looked Loop 27
phasenverriegelte Schleife 27
PLL 26
PLL-Einstellung 104
PLL-Jitter 29, 91
PM-Merkzettel 162
PRCHA 170
Printer Monitor . . . . . 139ff.
PRNUMB 167
Programmbibliothek 9
Programmnummer
  – ID 83
  – ØØ 94
  – FF 94
PROM 23

RAM/EPROM-Speicherkarte 68
RECCHA 173
RS 232 137
  – Interface 10
  – Peripherie 29

Schleifenfilter 28
SHOWPR 175
Speichererweiterung 18
  magnetische - 80
Speicherkapazität 9
Startadreßbyte 83
Startbit 125
Step-by-Step 30

Subroutinen
DUMP 90, 95
DUMPT 90, 95
RDTAPE 91, 95
Suchen des Programms 91

TAPE MANAGEMENT 90, 94
Tasten
BRK 148
EDIT 100
G 148
GET 99
L 145
M, ", ", CR 146
P 145
PAR 95
R 145
RES 140
RUB OUT 140
S 147
SAVE 98
SEF 101
Ø . . . 9, A . . . F, SP, und "." . . . 142
"+" und "–" 144
Tiefpaß 28

UART 123, 127
Umsetzung in tonfrequente
Signale . . . . . 87ff.

VCO 27
VIA 15
Videointerface . . . . . 121, 133ff.

Viele verbinden mit dem Stichwort "Computer" die Vorstellung: weniger Arbeit – mehr Freizeit. Aktueller und realistischer ist da schon die Formel: mehr Arbeit in gleicher Zeit. Die steigende Zahl der µComputer-Fans läßt noch einen anderen Schluß zu: mehr Arbeit in der Frei-Zeit!
Vor dem ersten Tastendruck steht allerdings die Qual der Wahl, denn das Angebot an Fix-und-fertig- oder Selbstbau-Computern, an Büchern und Zeitschriften ist sehr groß. Es soll schon Leute gegeben haben, die den Wald vor lauter Bäumen nicht sahen; dann entweder gar nicht anfingen oder zu Hause feststellen, daß der neue Home-Computer wohl doch ein Mißgriff war.

Elektor möchte deshalb allen Newcomern auf diesem Gebiet unter die Arme greifen:
- Mit dem Junior-Computer wird in Buch 1 ein preisgünstiger Ein-Platinen-Selbstbau-Mikrocomputer als Lernsystem angeboten.
- In Buch 2 ist die Rede von Monitor, Editor, Assemblieren und Disassemblieren.

Mit der in Buch 3 beschriebenen Interface-Karte wird aus dem ursprünglichen Ein-Platinen-Mikrocomputer ein ausgereiftes Mikrocomputersystem. Die Interfacekarte ist das Bindeglied zwischen dem Prozessor und dem Elekterminal (Datensichtgerät mit ASCII-Tastatur) sowie dem Arbeitsspeicher (RAM) und zwei Kassettenrekordern. Ferner enthält die Interface-Karte einen zusätzlichen I/O-Baustein, zwei RAM- und zwei EPROM-ICs sowie die Buspuffer, die für eine Erweiterung der Speicherkapazität über 8 K hinaus notwendig sind.
Neben der genauen Schaltungsbeschreibung für die Interface-Karte beschäftigt sich das Buch auch mit dem richtigen Umgang und der Bedienung für die Erweiterungen.
Das System ist also soweit ausbaufähig, wie es die spezielle oder allgemeine Anwendung fordert.
Also dann: Frisch an's Werk.

Elektor Verlag GmbH, D-5133 Gangelt 1

ISBN 3-921608-24-4